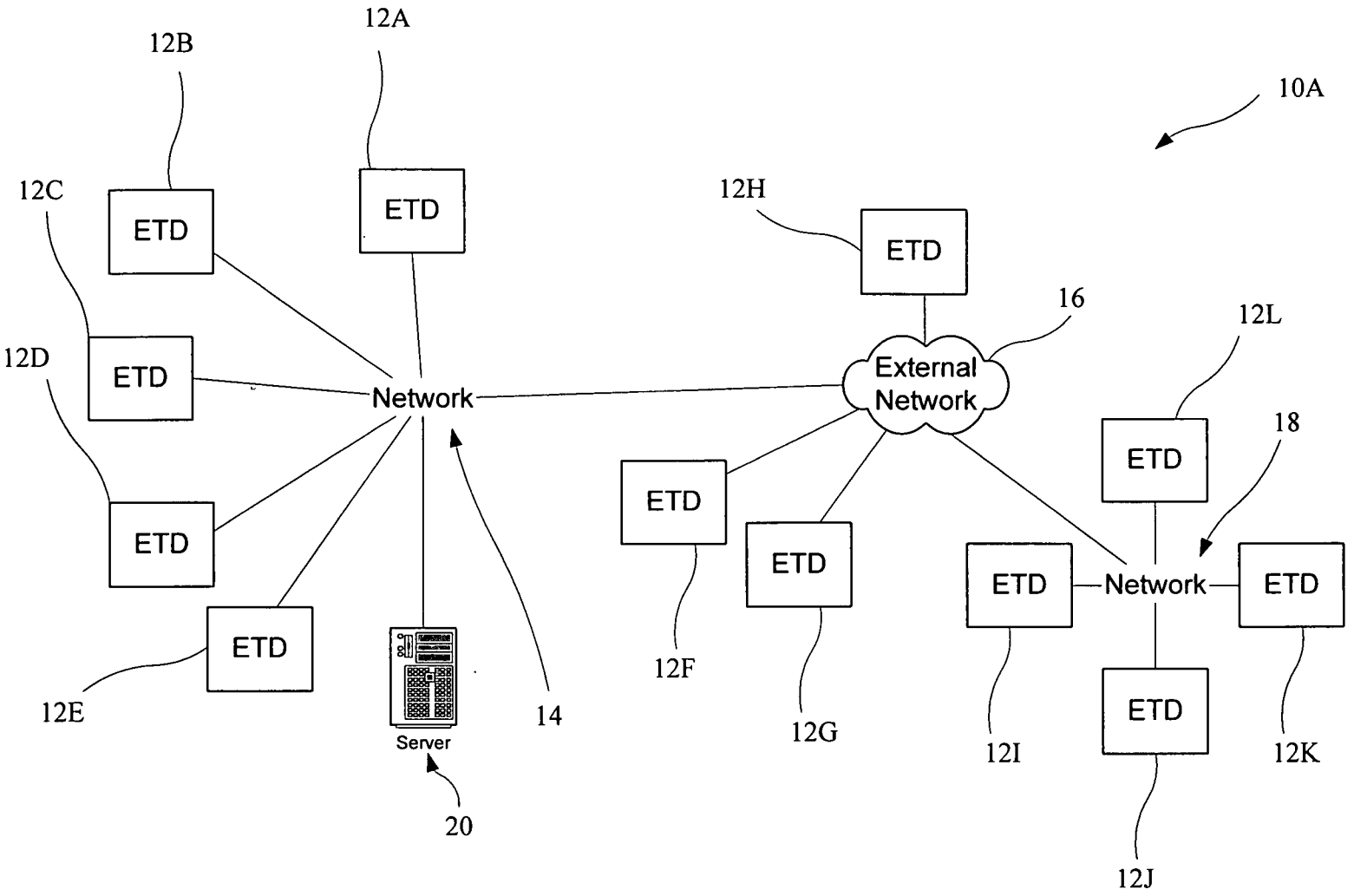

12A
12B
10A
12C
ETD
ETD
12H
ETD
16
12L
12D
ETD
Network
External
Network
ETD
18
ETD
ETD
ETD
ETD
Network
ETD
ETD
12F
ETD
12G
12E
Server
14
12I
12K
20
12J

FIG. 1A

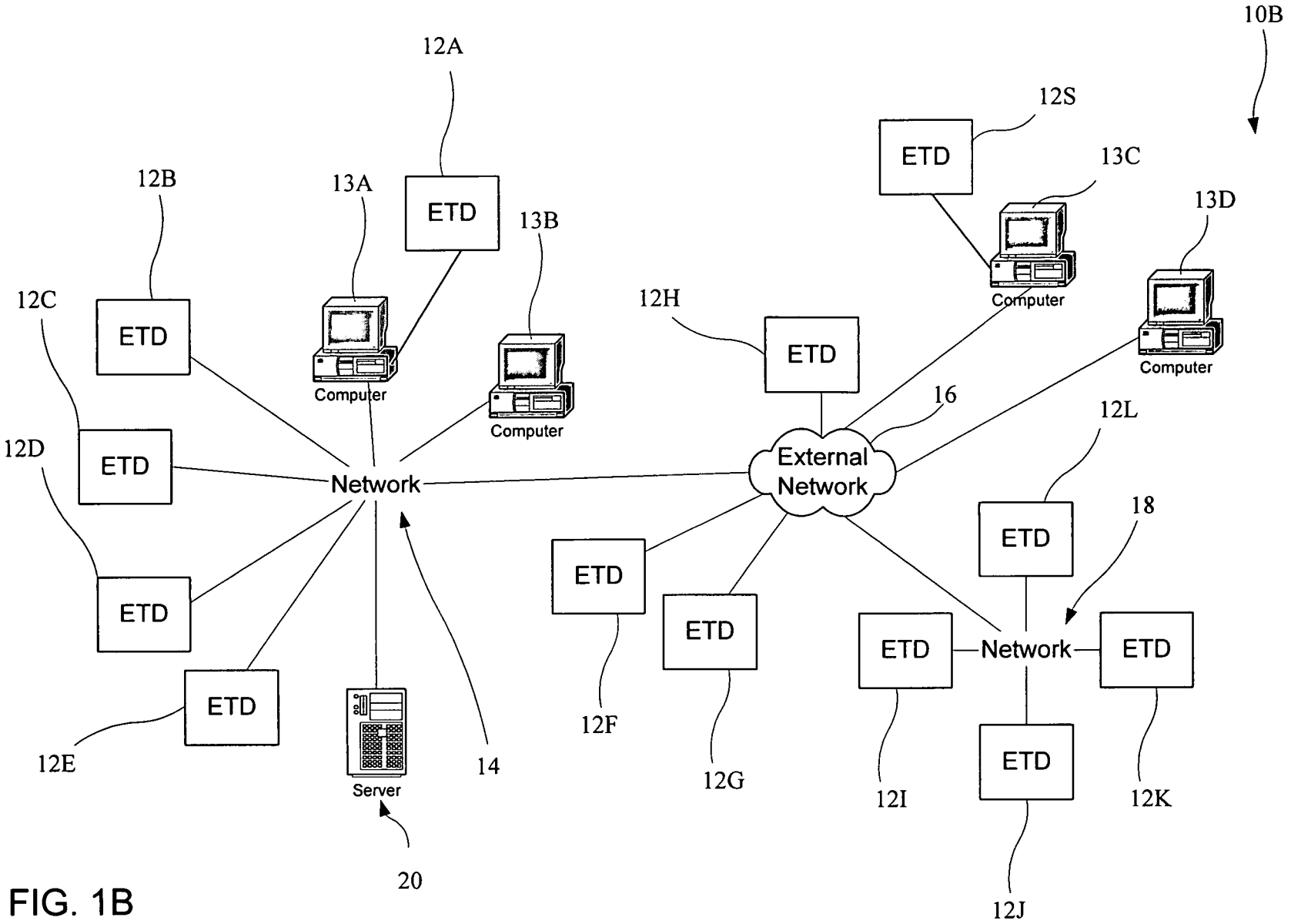

10B
12A
12S
ETD
13C
13D
12B
13A
ETD
13B
12C
ETD
Computer
Computer
12H
ETD
Computer
16
12L
12D
ETD
Network
External Network
ETD
18
ETD
ETD
ETD
Network
ETD
ETD
12E
Server
14
12F
12G
12I
ETD
12K
12J
20

FIG. 1B

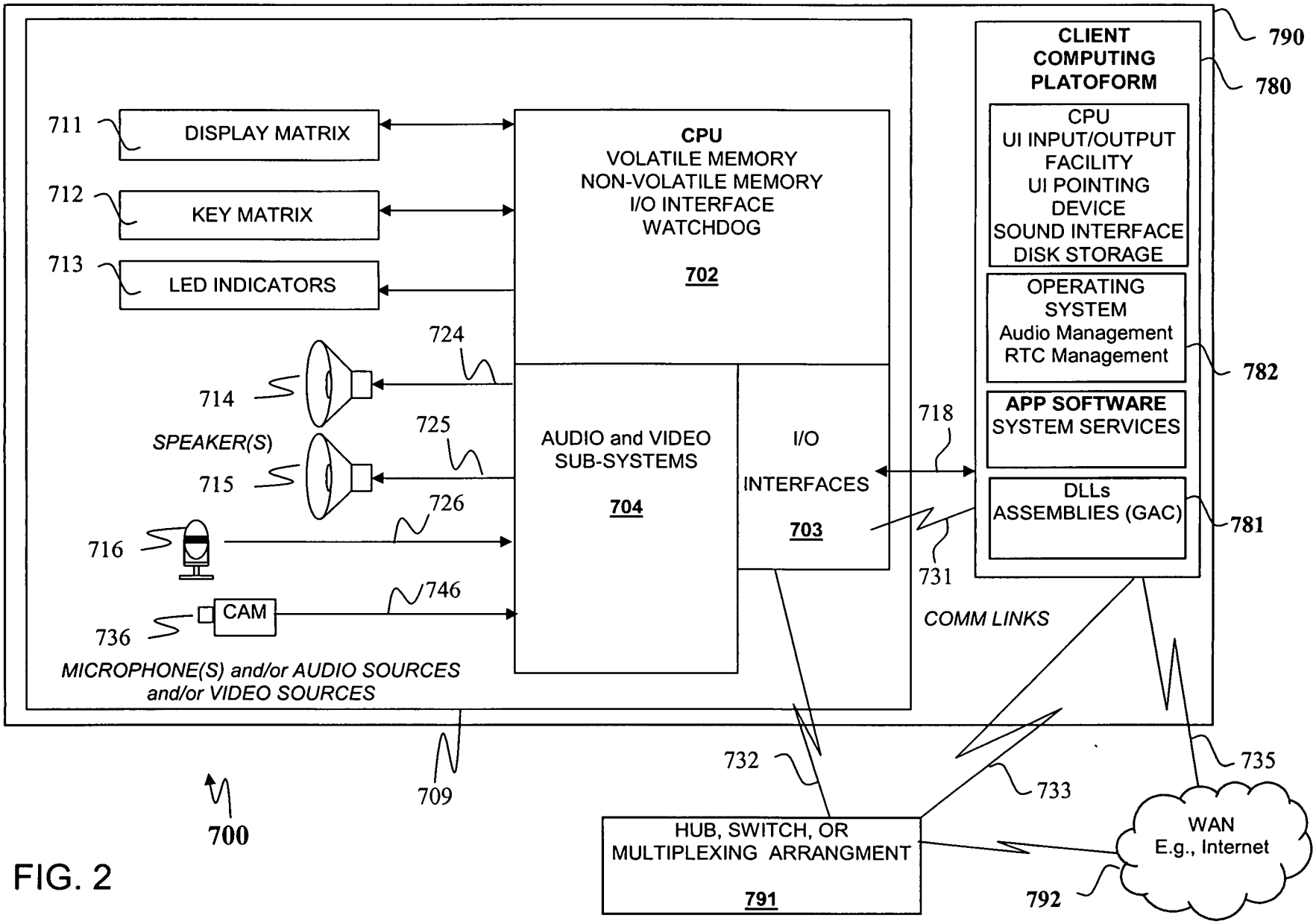

711
DISPLAY MATRIX
712
KEY MATRIX
713
LED INDICATORS
CPU
VOLATILE MEMORY
NON-VOLATILE MEMORY
I/O INTERFACE
WATCHDOG
702
724
714
SPEAKER(S)
725
715
726
716
746
CAM
736
MICROPHONE(S) and/or AUDIO SOURCES
and/or VIDEO SOURCES
AUDIO and VIDEO
SUB-SYSTEMS
704
I/O
INTERFACES
703
718
731
CLIENT
COMPUTING
PLATOFORM
790
780
CPU
UI INPUT/OUTPUT
FACILITY
UI POINTING
DEVICE
SOUND INTERFACE
DISK STORAGE
OPERATING
SYSTEM
Audio Management
RTC Management
782
APP SOFTWARE
SYSTEM SERVICES
DLLs
ASSEMBLIES (GAC)
781
COMM LINKS
709
700
732
733
735
HUB, SWITCH, OR
MULTIPLEXING ARRANGMENT
791
WAN
E.g., Internet
792

FIG. 2

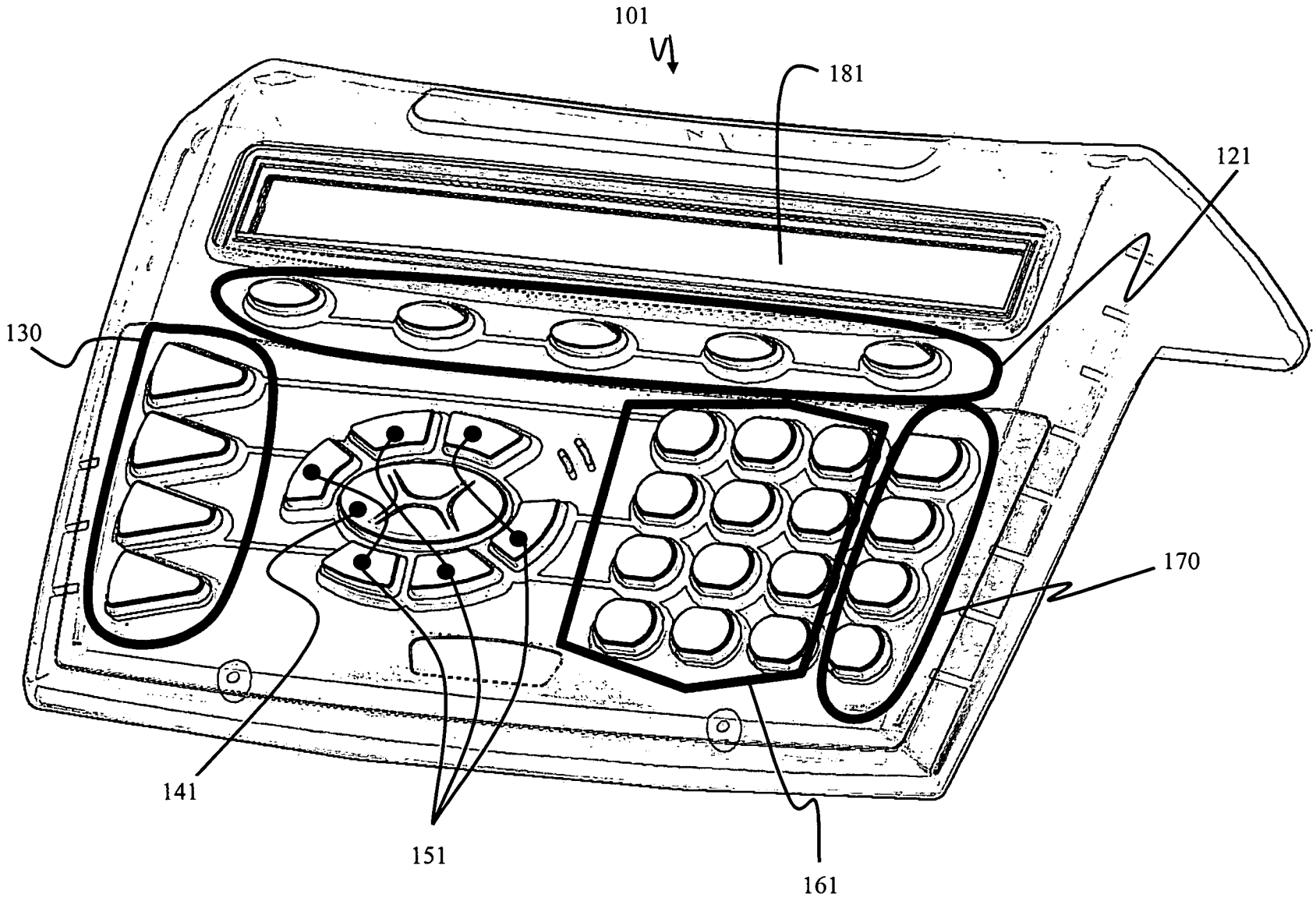


FIG. 3

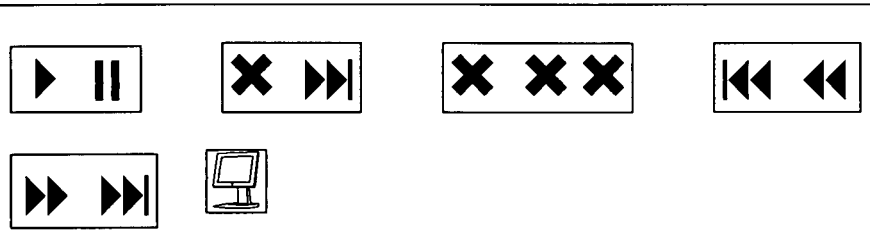
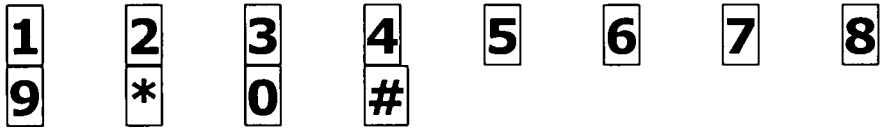
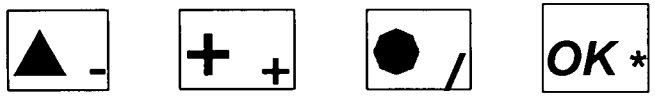
Graphic Representation	Text Representation (Definition)
142	• Function keys possibly used with a traditional soft-menu. • Group 121 buttons
143	• Predetermined semantic actions A, B, C and D; much like dedicated macro keys to carry out a set of one, or more tasks. • Grouped 130 buttons
144	• Cursor scroll keys for down, up, left and right • Group 141 buttons
145 	• Play (Stream) // Pause • Delete Current and Advance to Next • Delete Current // Delete All • Rewind to Beginning of Current // Rewind • Fast-Forward • Advance to End of Current • Browser or Application Summons • Group 151 buttons
146 	• Standard DTMF keys such as those found on a Touch-Tone® keypad. • Group 161 buttons
147 	• Originate // Insert (Subtract) • Add-To Semantic (Addition) • Record / Record-Stop (Division) • Okay-Accept (Multiplication) • Group 170 buttons

FIG. 4

140

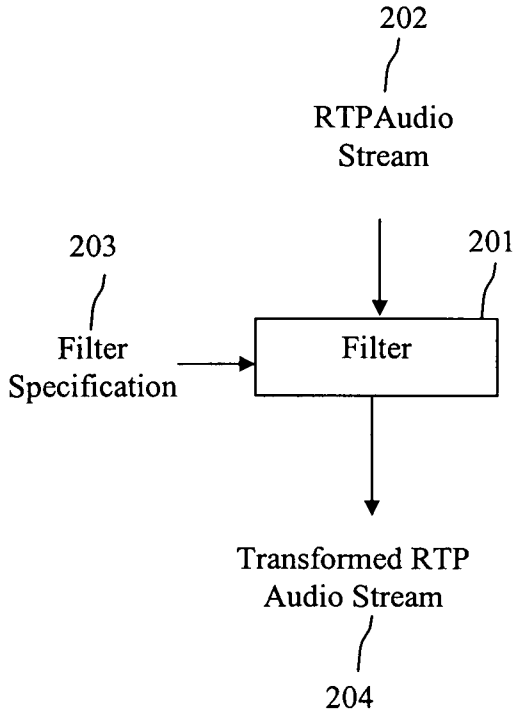

202
RTPAudio
Stream
203
201
Filter
Specification
Filter
Transformed RTP
Audio Stream
204

FIG. 5A

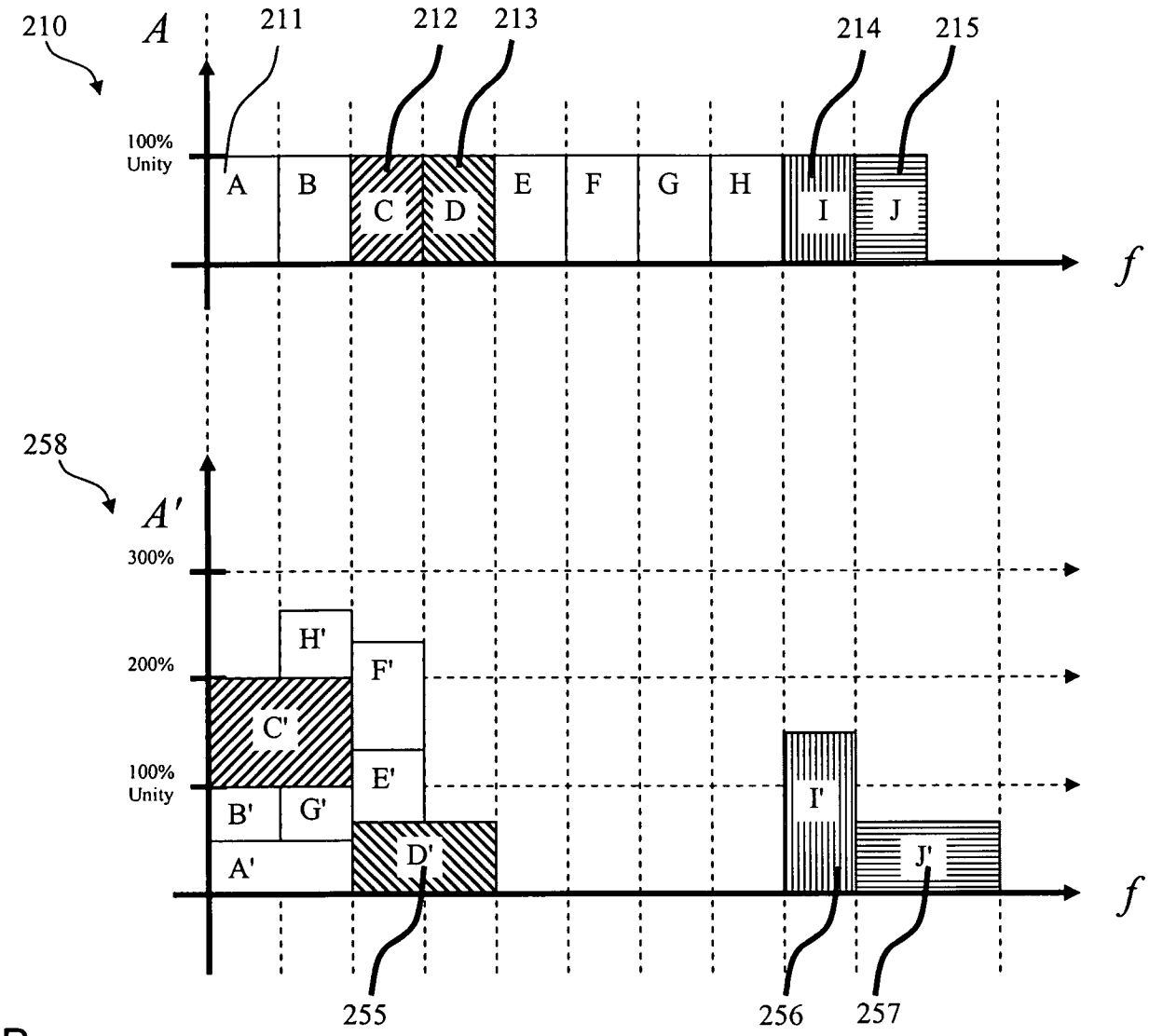

210
A
211
212
213
214
215
100%
Unity
A
B
C
D
E
F
G
H
I
J
f
258
A'
300%
200%
100%
Unity
H'
F'
C'
E'
B'
G'
A'
D'
I'
J'
f
255
256
257
200

FIG. 5B

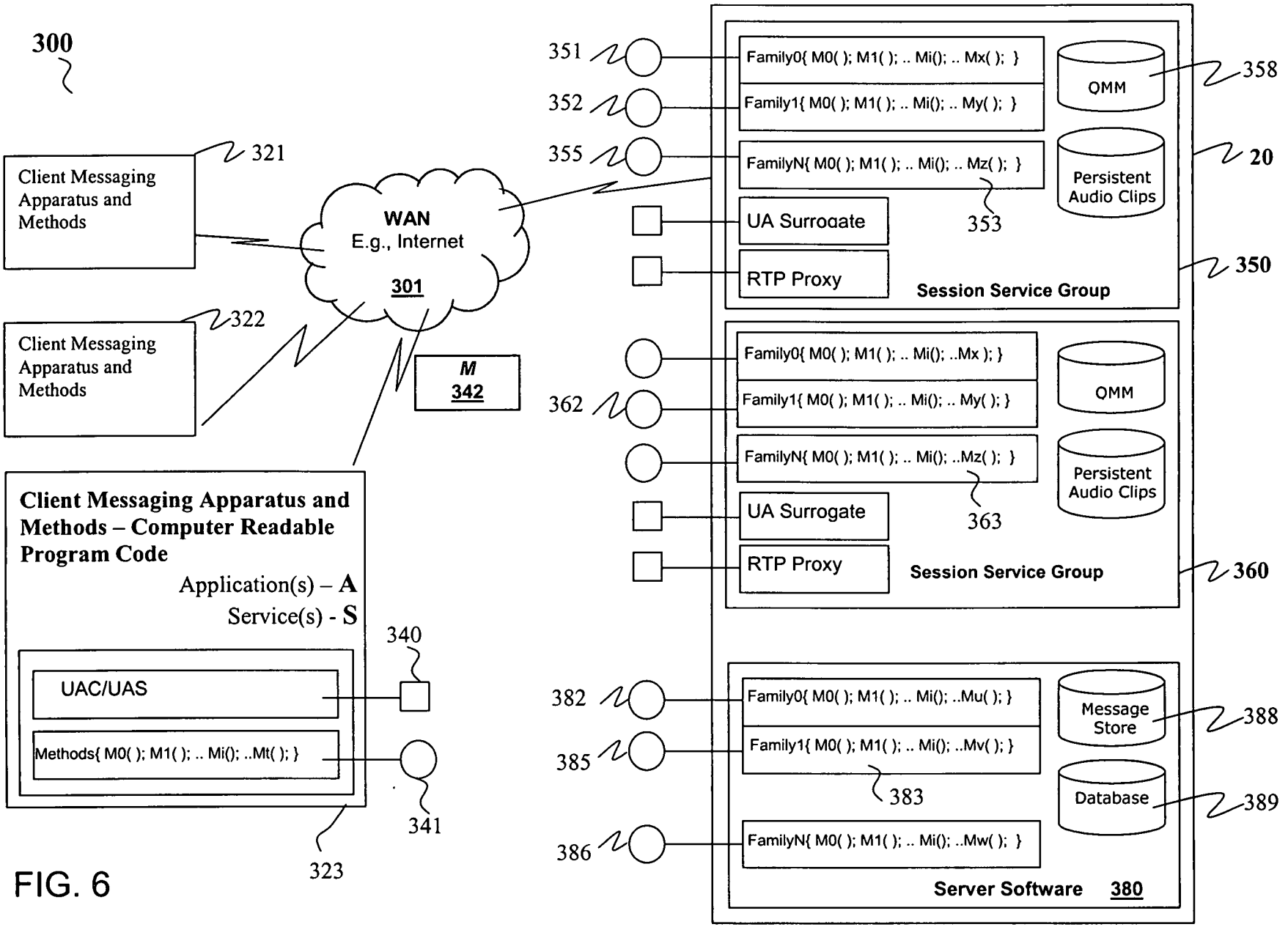

300
Client Messaging Apparatus and Methods
321
WAN
E.g., Internet
301
Client Messaging Apparatus and Methods
322
M
342
Client Messaging Apparatus and Methods – Computer Readable Program Code
Application(s) – A
Service(s) - S
UAC/UAS
Methods{ M0( ); M1( ); .. Mi(); ..Mt( ); }
340
341
323
351
352
355
Family0{ M0( ); M1( ); .. Mi(); .. Mx( ); }
Family1{ M0( ); M1( ); .. Mi(); .. My( ); }
FamilyN{ M0( ); M1( ); .. Mi(); .. Mz( ); }
353
UA Surrogate
RTP Proxy
Session Service Group
OMM
Persistent Audio Clips
358
20
350
362
Family0{ M0( ); M1( ); .. Mi(); ..Mx); }
Family1{ M0( ); M1( ); .. Mi(); .. My( ); }
FamilyN{ M0( ); M1( ); .. Mi(); ..Mz( ); }
363
UA Surrogate
RTP Proxy
Session Service Group
OMM
Persistent Audio Clips
360
382
385
386
Family0{ M0( ); M1( ); .. Mi(); ..Mu( ); }
Family1{ M0( ); M1( ); .. Mi(); ..Mv( ); }
383
FamilyN{ M0( ); M1( ); .. Mi(); ..Mw( ); }
Server Software 380
Message Store
388
Database
389

FIG. 6

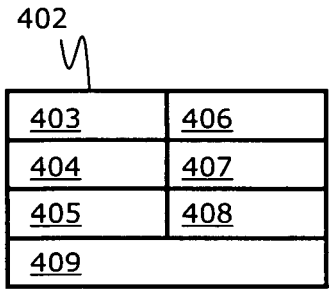


FIG. 7A

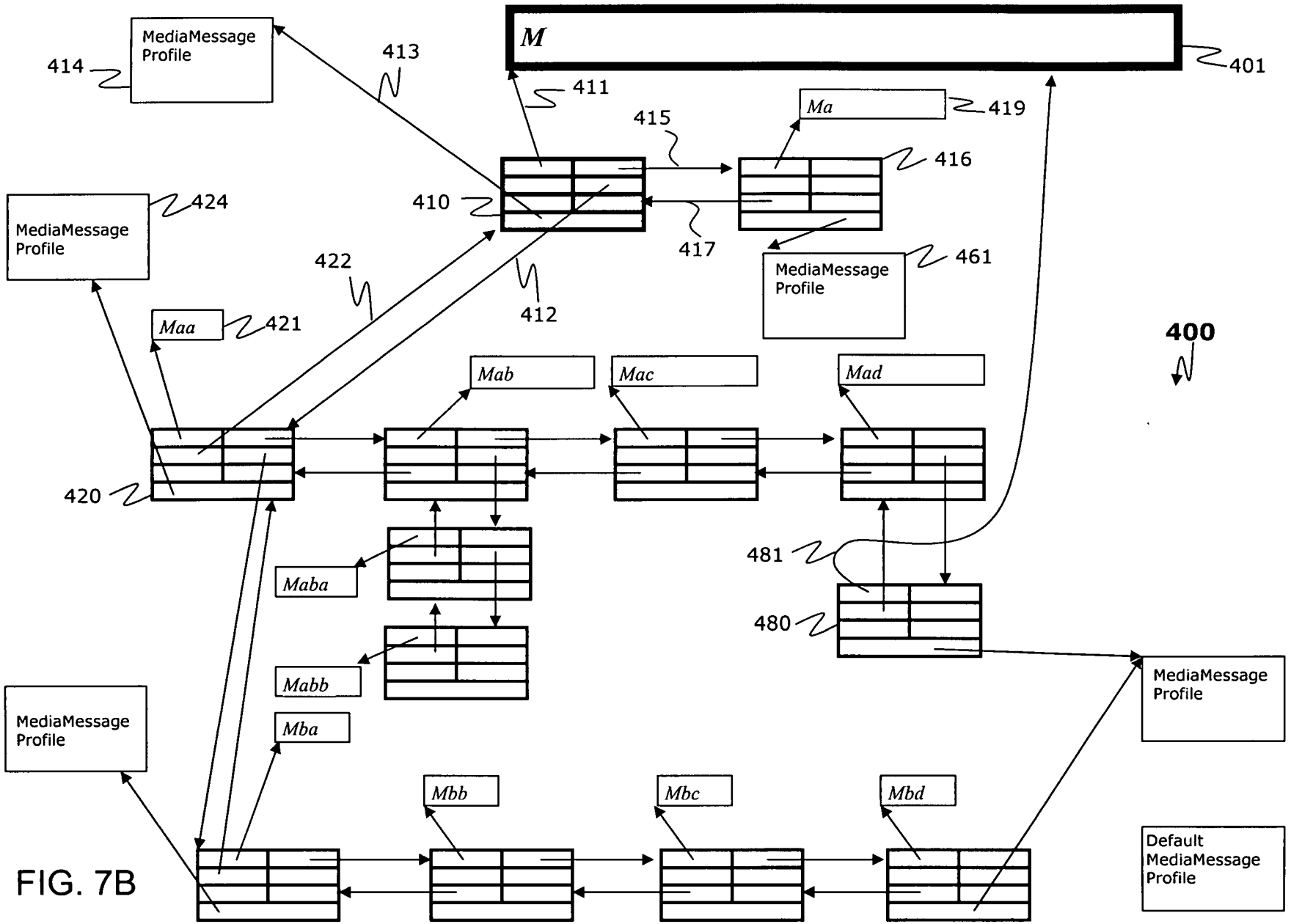

M
401
MediaMessage Profile
414
413
411
Ma
419
415
416
410
417
MediaMessage Profile
424
MediaMessage Profile
461
422
412
Maa
421
400
Mab
Mac
Mad
420
Maba
481
480
Mabb
MediaMessage Profile
MediaMessage Profile
Mba
Mbb
Mbc
Mbd
Default MediaMessage Profile

FIG. 7B

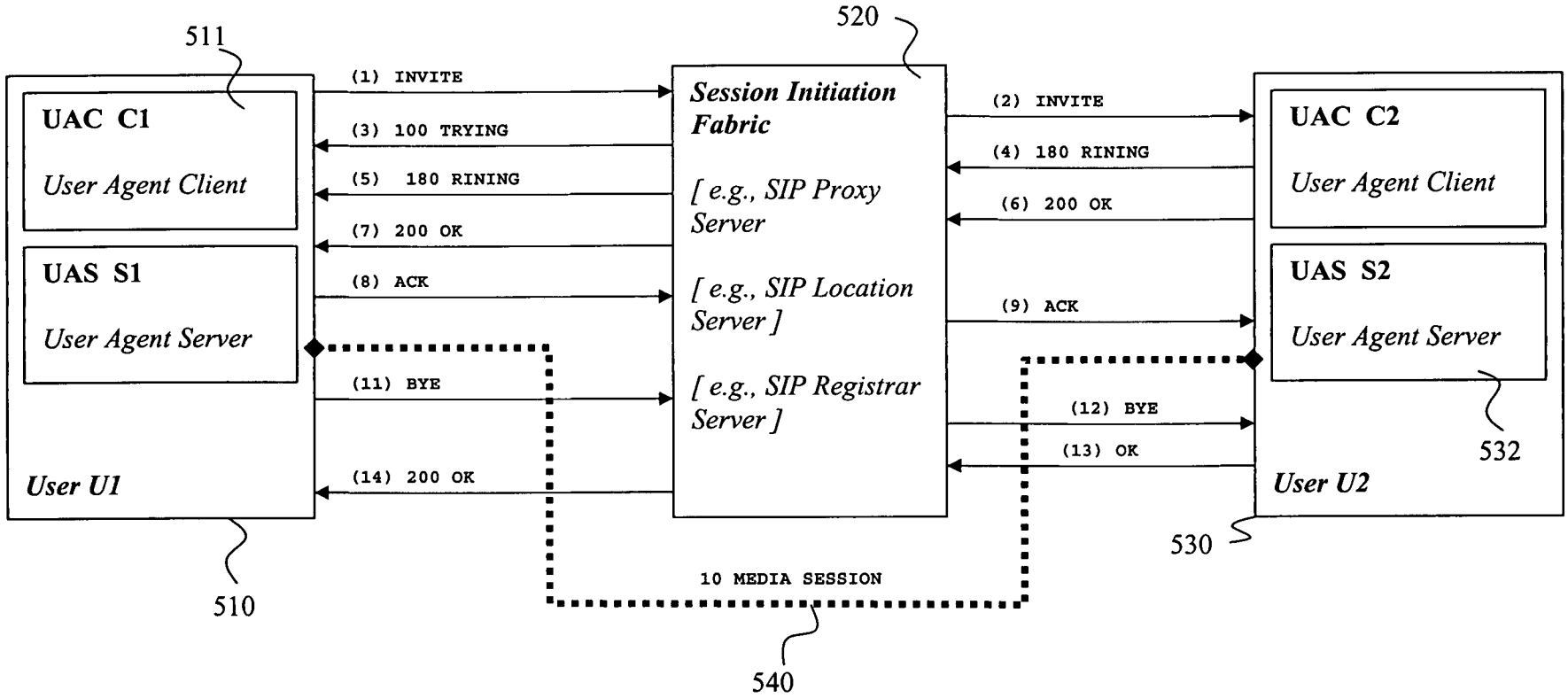


FIG. 8A **PRIOR ART**

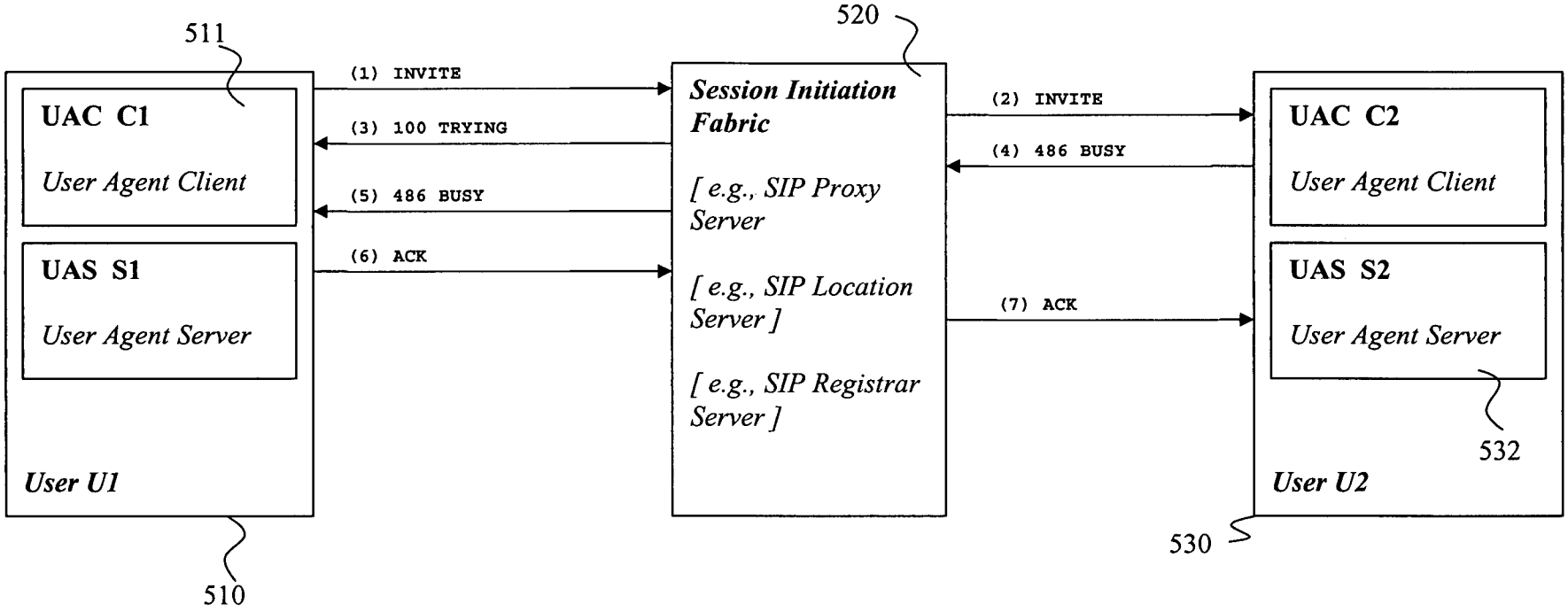


FIG. 8B **PRIOR ART**

```
<?xml version="1.0" encoding="utf-8" ?>
<SchemaRoot>
     <MediaMessageProfile><!600>
          <MediaMessageObjectURI></MediaMessageObjectURI><!611>
          <MediaMessageDescriptor></MediaMessageDescriptor><!612>
          <TargetMessageScrollTimeMilliSeconds></TargetMessageScrollTimeMilliSeconds><!613>
          <MessageOffsetInMilliSeconds></MessageOffsetInMilleSeconds><!614>
          <MessageDurationInMilliSeconds></MessageDurationInMilliSeconds><!615>
          <MessageMarkUpType></MessageMarkUpType><!616>
          <MaxNumberOfChildren></MaxNumberOfChildren><!617>
          <MaxNumberOfPeers></MaxNumberOfPeers><!618>
          <ProfileCreationTimeSecondsFromEpoch></ProfileCreationTimeSecondsFromEpoch><!619>
          <ProfileLifetimeInSeconds></ProfileLifetimeInSeconds><!620>
          <ObjectCreationTimeSecondsFromEpoch></ObjectCreationTimeSecondsFromEpoch><!621>
          <ObjectLifetimeInSeconds></ObjectLifetimeInSeconds><!622>
          <OwnerURI></OwnerURI><!623>
          <FilterProfileURI></FilterProfileURI><!624>
          <AuthorizationAttributes></AuthorizationAttributes><!625>
          <AuthenticationAttributes></AuthenticationAttributes><!626>
          <AdditionalInformation></AdditionalInformation><!627>
     </MediaMessageProfile>
</SchemaRoot>
```

FIG. 9A

640

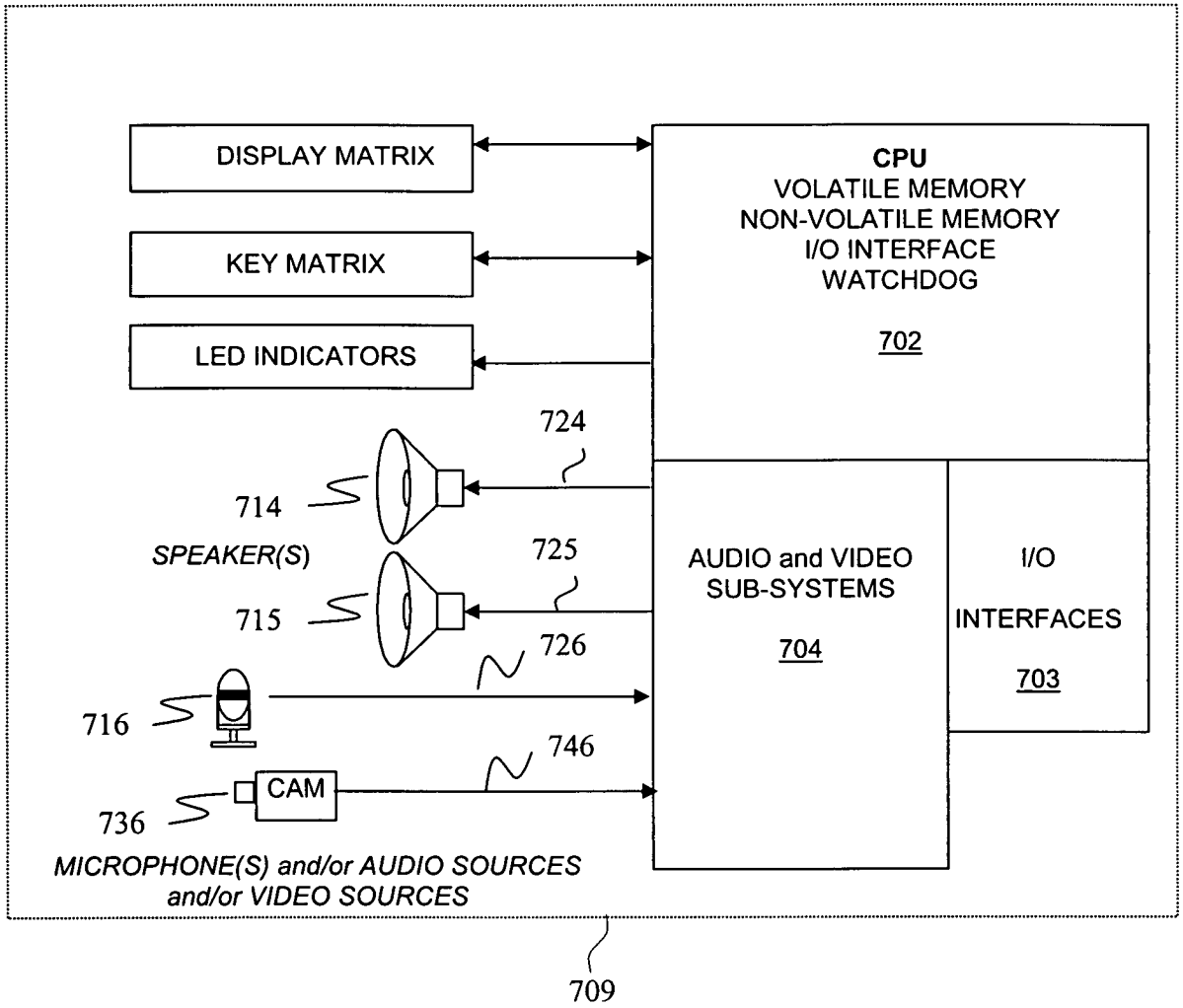
```
<?xml version="1.0" encoding="utf-8" ?>
<SchemaRoot>
      <MediaMessageObject><!640>
            <MediaMessageContentURI><!411></MediaMessageContentURI><!643>
            <ParentMediaMessageURI><!422></ParentMediaMessageURI><!644>
            <MediaMessageReversePeerURI><!417></MediaMessageReversePeerURI><!645>
            <MediaMessageForwardPeerURI><!415></MediaMessageForwardPeerURI><!646>
            <MediaMessageForwardChildURI><!412></MediaMessageForwardChildURI><!647>
            <OversightWebServiceURI><!648></OversightWebServiceURI><!648>
            <MediaMessageProfileURI><!413></MediaMesasgeProfileURI><!649>
            <ObjectCreationTimeSecondsFromEpoch></ObjectCreationTimeSecondsFromEpoch><!650>
            <ObjectLifetimeInSeconds></ObjectLifetimeInSeconds><!651>
            <OwnerURI></OwnerURI><!652>
            <AuthorizationAttributes></AuthorizationAttributes><!653>
            <AuthenticationAttributes></AuthenticationAttributes><!654>
            <AdditionalInformation></AdditionalInformation><!655>
      </MediaMessageObject>
</SchemaRoot>
```

FIG. 9B

680

```
<?xml version="1.0" encoding="utf-8" ?>
<SchemaRootFilterProfile>
      <FilterProfile><!680>
            <NumberOfBands></NumberOfBands><!681>
            <BandDefinitions><!682>
                  <Band><!683>
                        <StartHz></StartHz><!684>
                        <EndHz></EndHz><!685>
                        <TranslationStartHz></TranslationStartHz><!686>
                        <TranslationEndHz></TranslationEndHz><!687>
                        <AmplitideCoeifficent></AmplitideCoeifficent><!688>
                  </Band>
                  <Band>
                        <StartHz></StartHz>
                        <EndHz></EndHz>
                        <TranslationStartHz></TranslationStartHz>
                        <TranslationEndHz></TranslationEndHz>
                        <AmplitideCoeifficent></AmplitideCoeifficent>
                  </Band>
            </BandDefinitions>
      </FilterProfie>
</SchemaRootFilterProfile>
```

FIG. 9C


DISPLAY MATRIX
KEY MATRIX
LED INDICATORS
CPU
VOLATILE MEMORY
NON-VOLATILE MEMORY
I/O INTERFACE
WATCHDOG
702
724
714
SPEAKER(S)
725
715
726
716
746
CAM
736
MICROPHONE(S) and/or AUDIO SOURCES
and/or VIDEO SOURCES
AUDIO and VIDEO
SUB-SYSTEMS
704
I/O
INTERFACES
703
709

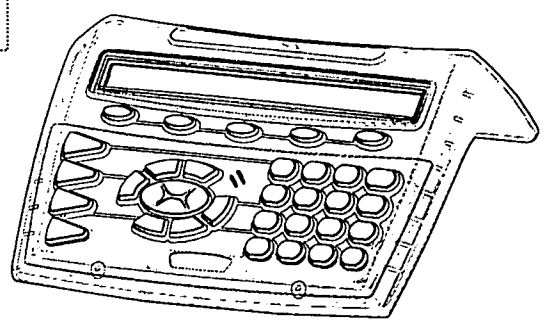


FIG. 10

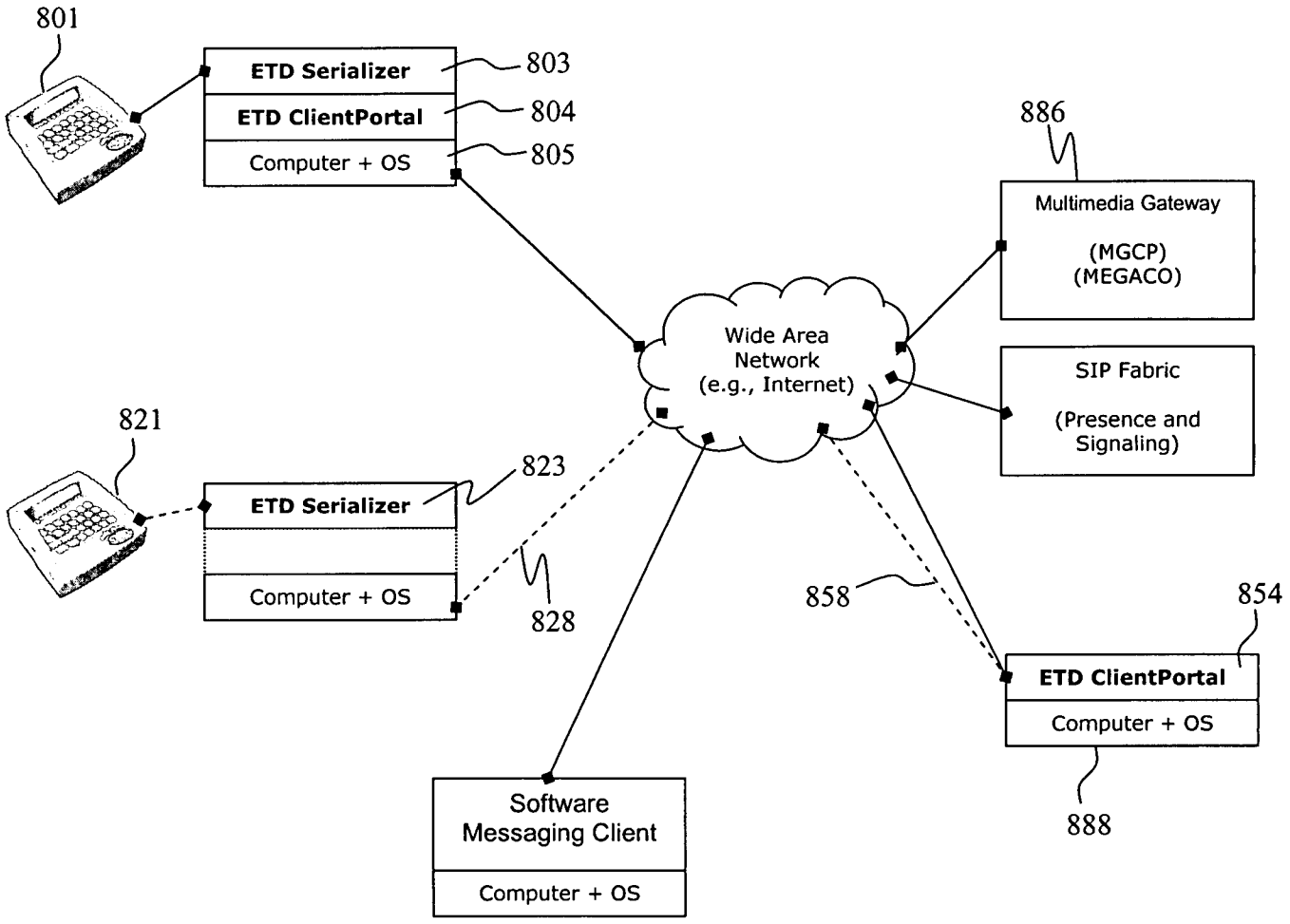

801
ETD Serializer
803
ETD ClientPortal
804
Computer + OS
805
886
Multimedia Gateway
(MGCP)
(MEGACO)
Wide Area
Network
(e.g., Internet)
SIP Fabric
(Presence and
Signaling)
821
823
ETD Serializer
Computer + OS
828
858
854
ETD ClientPortal
Computer + OS
888
Software
Messaging Client
Computer + OS

FIG. 11

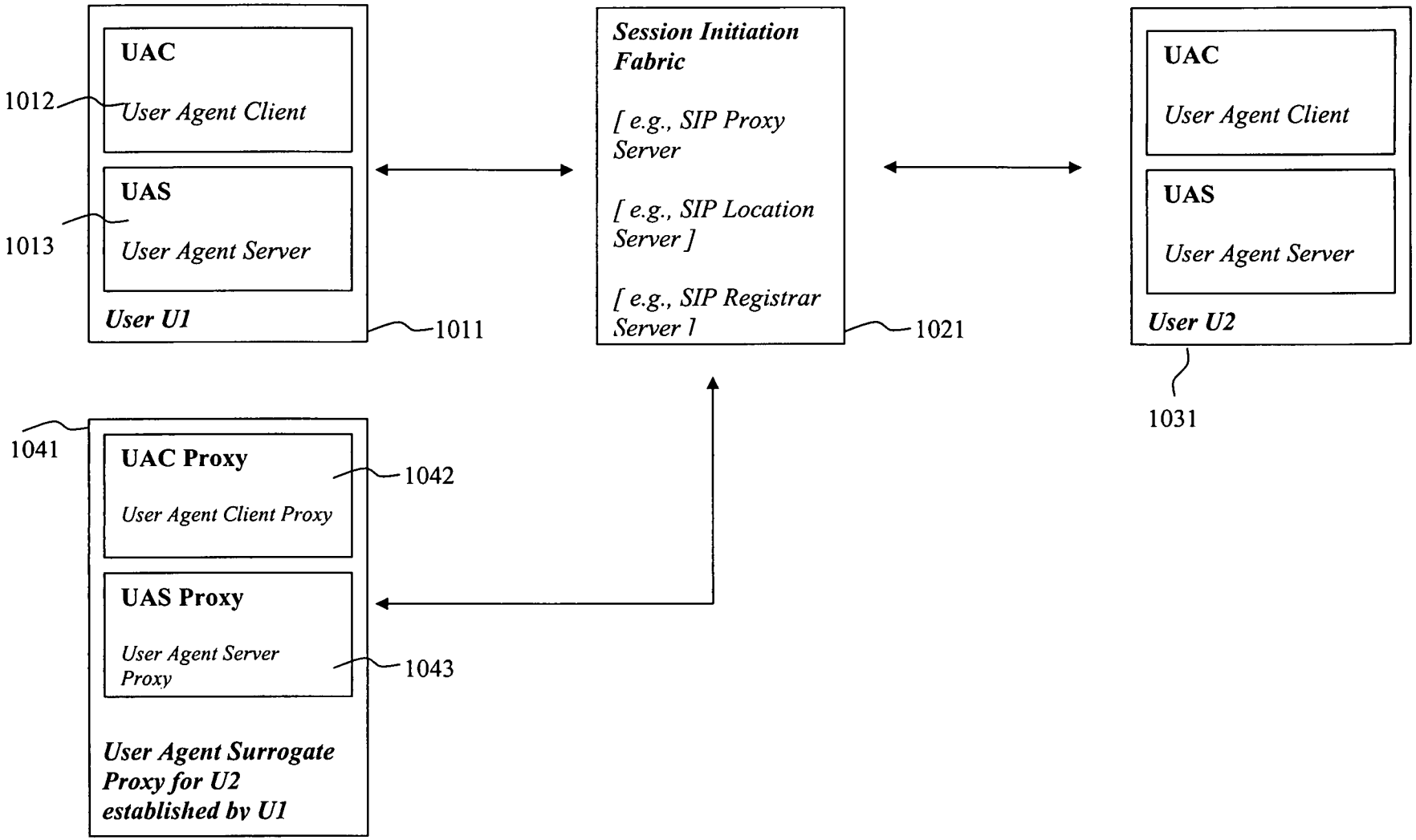

UAC
User Agent Client
1012
UAS
User Agent Server
1013
User U1
1011
Session Initiation Fabric
[e.g., SIP Proxy Server
[e.g., SIP Location Server]
[e.g., SIP Registrar Server]
1021
UAC
User Agent Client
UAS
User Agent Server
User U2
1031
1041
UAC Proxy
User Agent Client Proxy
1042
UAS Proxy
User Agent Server Proxy
1043
User Agent Surrogate Proxy for U2 established by U1

FIG. 12A

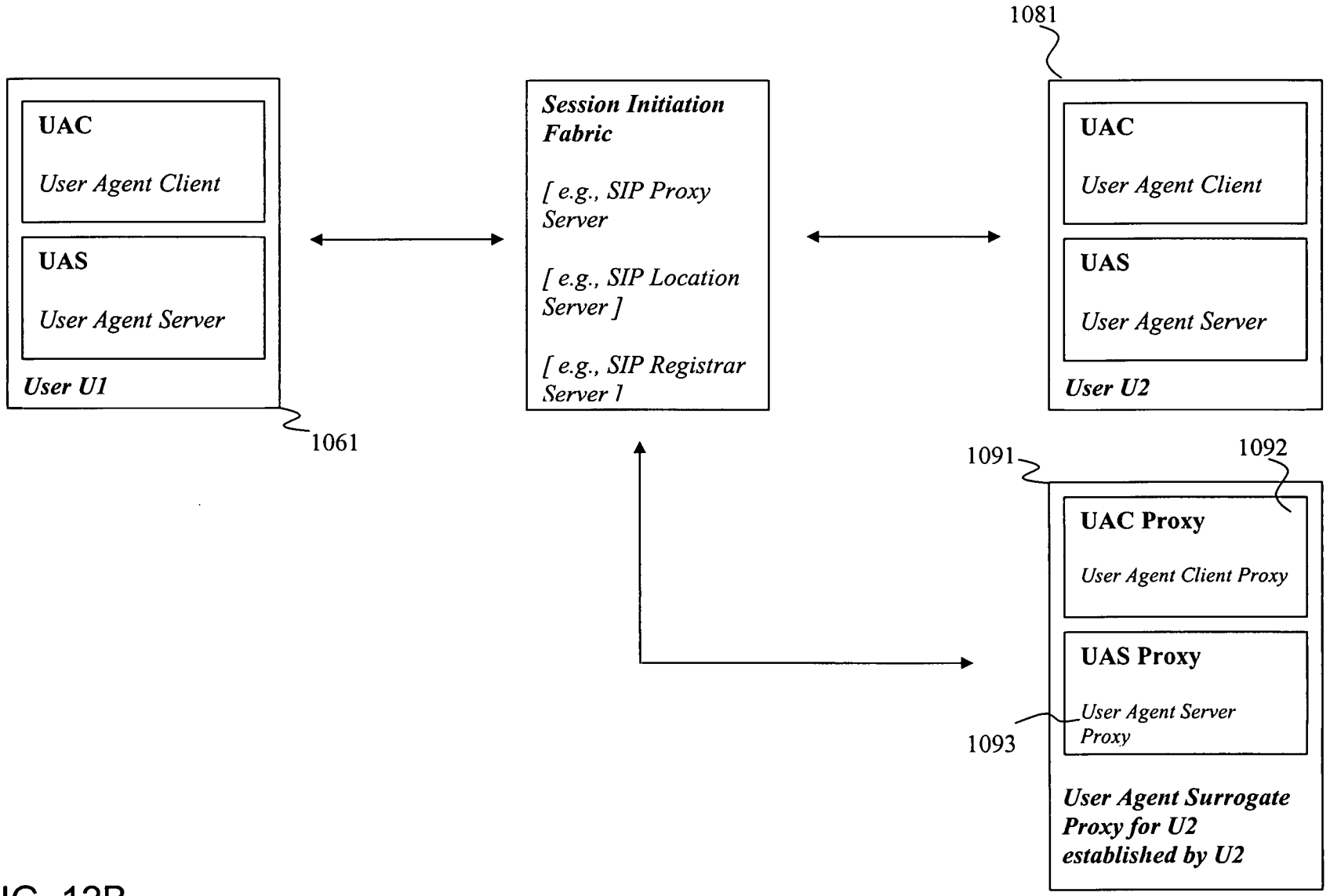

UAC
User Agent Client
UAS
User Agent Server
User U1
1061
Session Initiation Fabric
[e.g., SIP Proxy Server
[e.g., SIP Location Server]
[e.g., SIP Registrar Server]
1081
UAC
User Agent Client
UAS
User Agent Server
User U2
1091
1092
UAC Proxy
User Agent Client Proxy
UAS Proxy
User Agent Server Proxy
1093
User Agent Surrogate Proxy for U2 established by U2

FIG. 12B

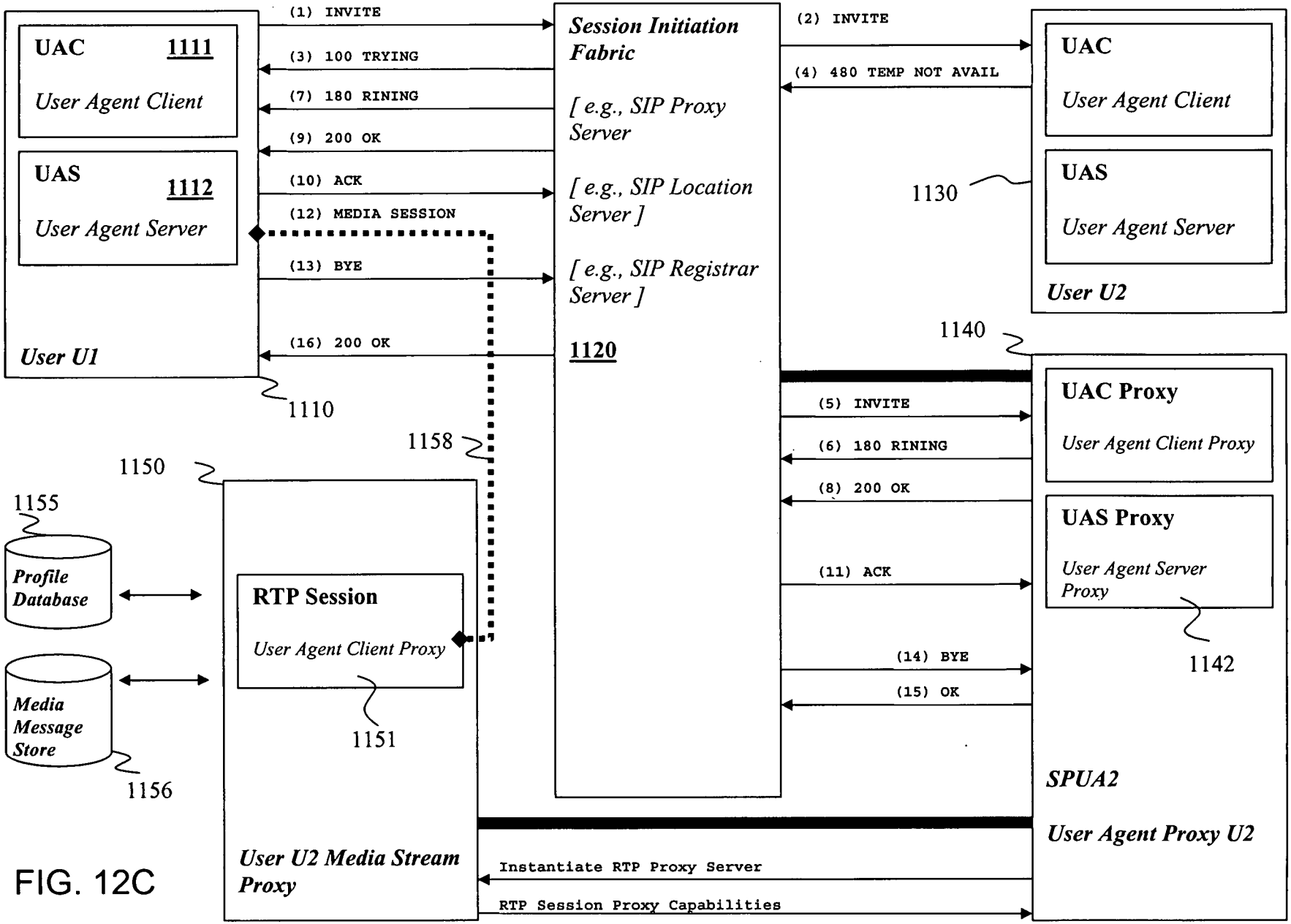

UAC 1111
User Agent Client
UAS 1112
User Agent Server
User U1
1110
(1) INVITE
(3) 100 TRYING
(7) 180 RINING
(9) 200 OK
(10) ACK
(12) MEDIA SESSION
(13) BYE
(16) 200 OK
Session Initiation Fabric
[e.g., SIP Proxy Server
[e.g., SIP Location Server]
[e.g., SIP Registrar Server]
1120
(2) INVITE
(4) 480 TEMP NOT AVAIL
1130
UAC
User Agent Client
UAS
User Agent Server
User U2
1140
UAC Proxy
User Agent Client Proxy
UAS Proxy
User Agent Server Proxy
1142
(5) INVITE
(6) 180 RINING
(8) 200 OK
(11) ACK
(14) BYE
(15) OK
SPUA2
User Agent Proxy U2
1158
1150
1155
Profile Database
Media Message Store
1156
RTP Session
User Agent Client Proxy
1151
User U2 Media Stream Proxy
Instantiate RTP Proxy Server
RTP Session Proxy Capabilities

FIG. 12C

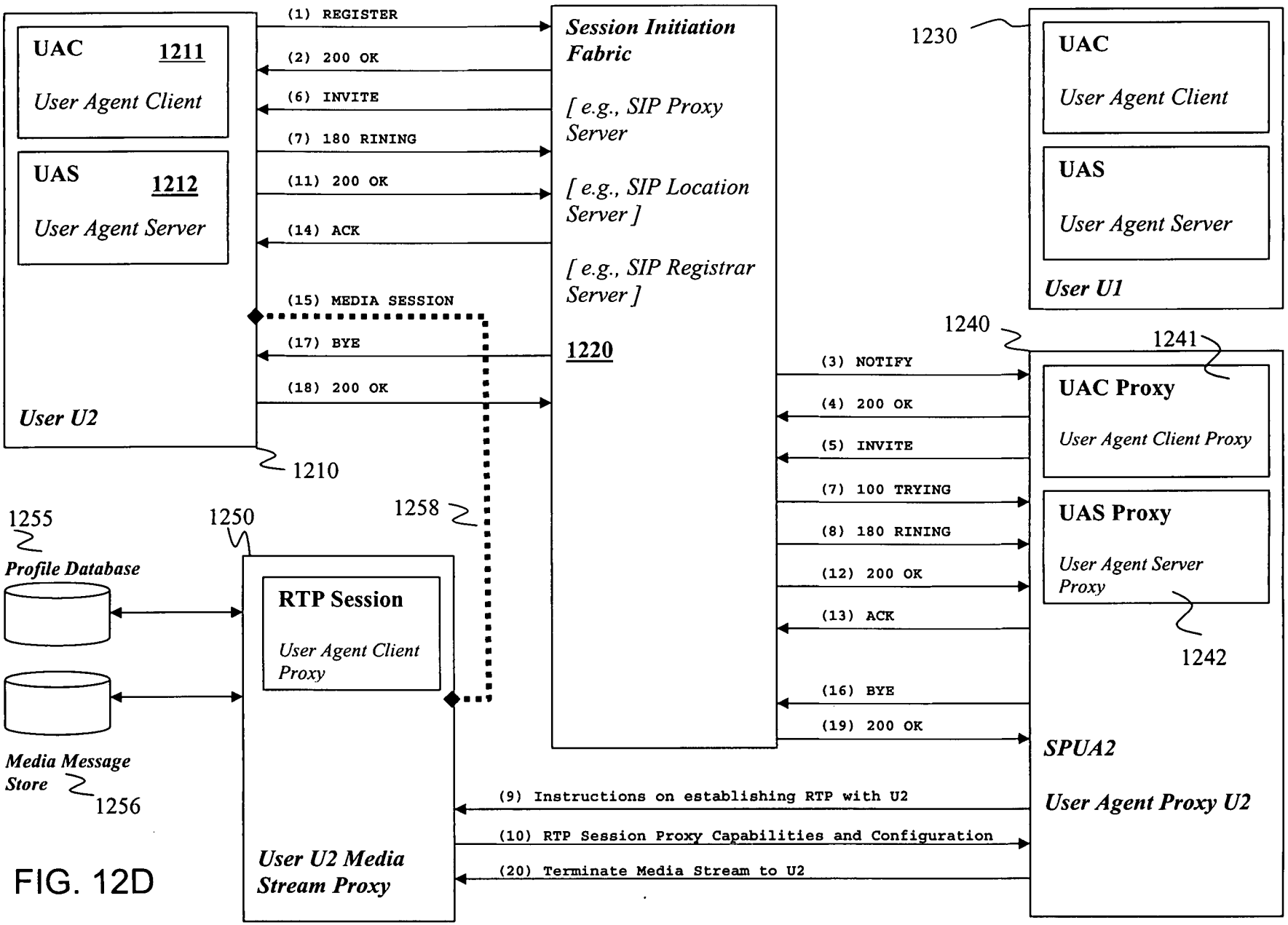

UAC 1211
User Agent Client
UAS 1212
User Agent Server
User U2
1210
(1) REGISTER
(2) 200 OK
(6) INVITE
(7) 180 RINING
(11) 200 OK
(14) ACK
(15) MEDIA SESSION
(17) BYE
(18) 200 OK
Session Initiation Fabric
[e.g., SIP Proxy Server
[e.g., SIP Location Server]
[e.g., SIP Registrar Server]
1220
1230
UAC
User Agent Client
UAS
User Agent Server
User U1
1240
1241
UAC Proxy
User Agent Client Proxy
UAS Proxy
User Agent Server Proxy
1242
SPUA2
User Agent Proxy U2
(3) NOTIFY
(4) 200 OK
(5) INVITE
(7) 100 TRYING
(8) 180 RINING
(12) 200 OK
(13) ACK
(16) BYE
(19) 200 OK
1255
Profile Database
Media Message Store
1256
1250
1258
RTP Session
User Agent Client Proxy
User U2 Media Stream Proxy
(9) Instructions on establishing RTP with U2
(10) RTP Session Proxy Capabilities and Configuration
(20) Terminate Media Stream to U2

FIG. 12D

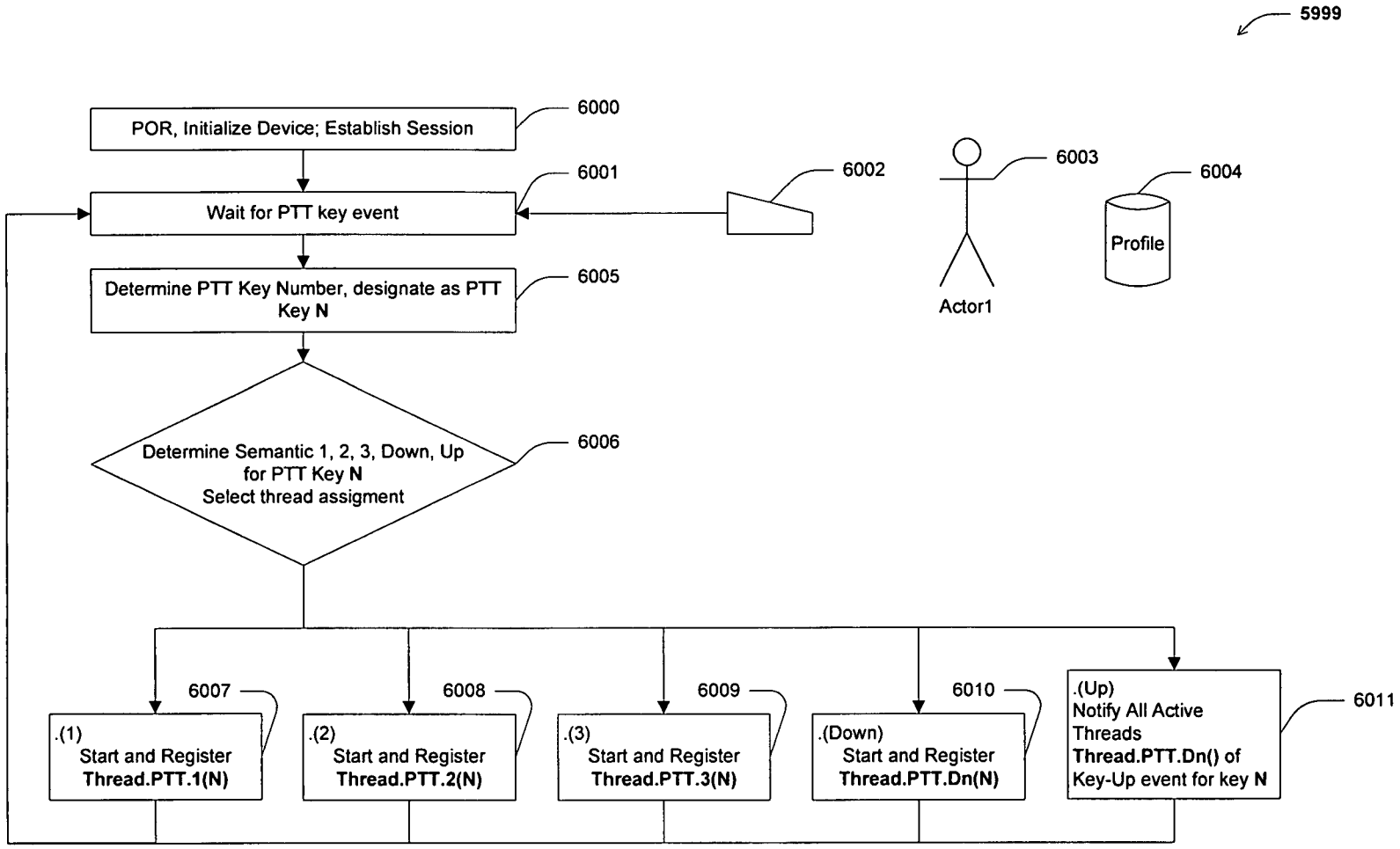

5999
POR, Initialize Device; Establish Session
6000
Wait for PTT key event
6001
6002
6003
Actor1
6004
Profile
Determine PTT Key Number, designate as PTT Key N
6005
Determine Semantic 1, 2, 3, Down, Up for PTT Key N
Select thread assigment
6006
6007
.(1)
Start and Register
Thread.PTT.1(N)
6008
.(2)
Start and Register
Thread.PTT.2(N)
6009
.(3)
Start and Register
Thread.PTT.3(N)
6010
.(Down)
Start and Register
Thread.PTT.Dn(N)
.(Up)
Notify All Active
Threads
Thread.PTT.Dn() of
Key-Up event for key N
6011

FIG. 13A

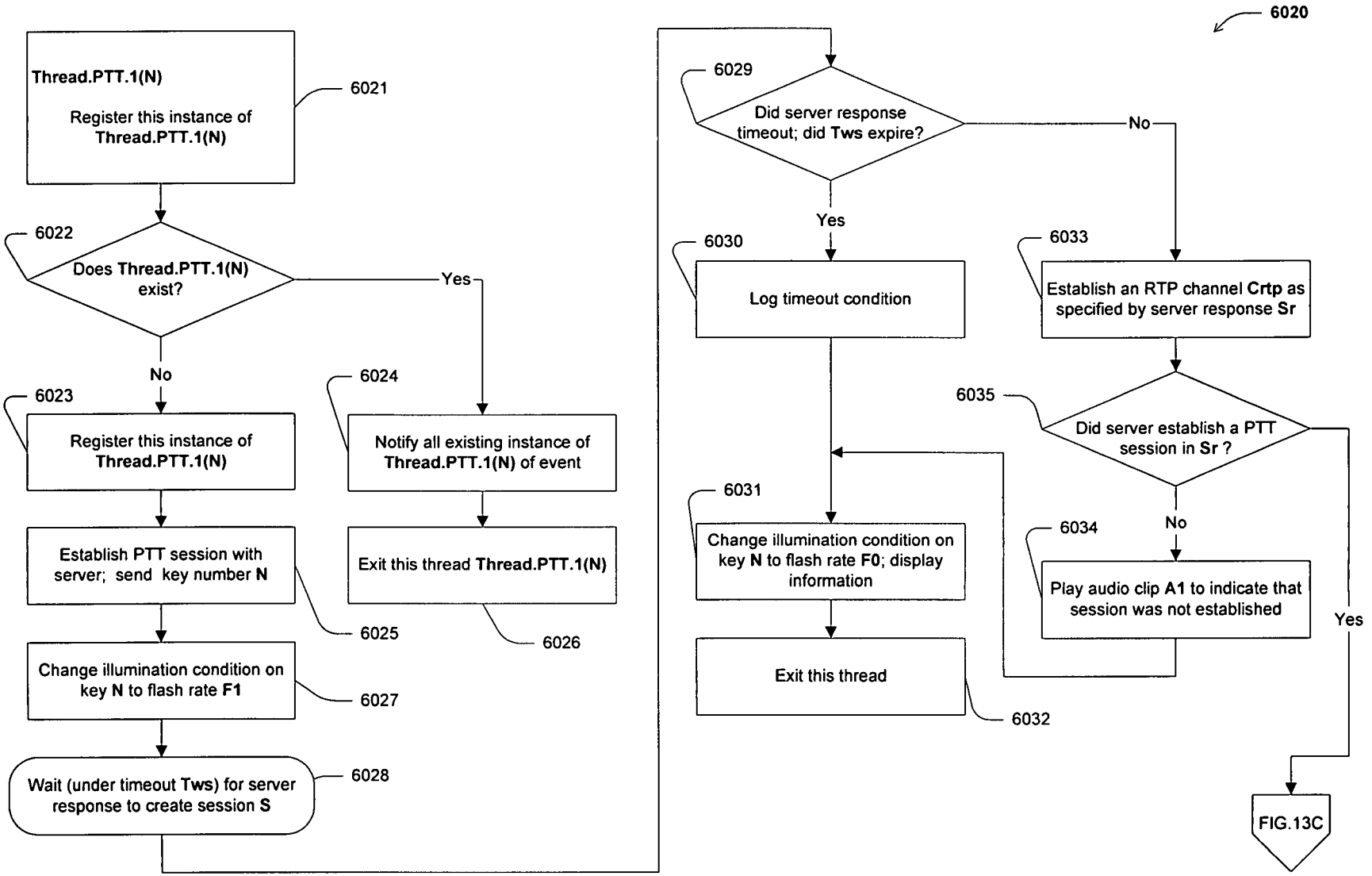

Thread.PTT.1(N)
Register this instance of Thread.PTT.1(N)
6021
6022
Does Thread.PTT.1(N) exist?
Yes
No
6023
Register this instance of Thread.PTT.1(N)
6024
Notify all existing instance of Thread.PTT.1(N) of event
Establish PTT session with server; send key number N
Exit this thread Thread.PTT.1(N)
6025
6026
Change illumination condition on key N to flash rate F1
6027
Wait (under timeout Tws) for server response to create session S
6028
6020
6029
Did server response timeout; did Tws expire?
No
Yes
6030
Log timeout condition
6033
Establish an RTP channel Crtp as specified by server response Sr
6035
Did server establish a PTT session in Sr ?
6031
Change illumination condition on key N to flash rate F0; display information
6034
No
Play audio clip A1 to indicate that session was not established
Yes
Exit this thread
6032
FIG.13C

FIG. 13B

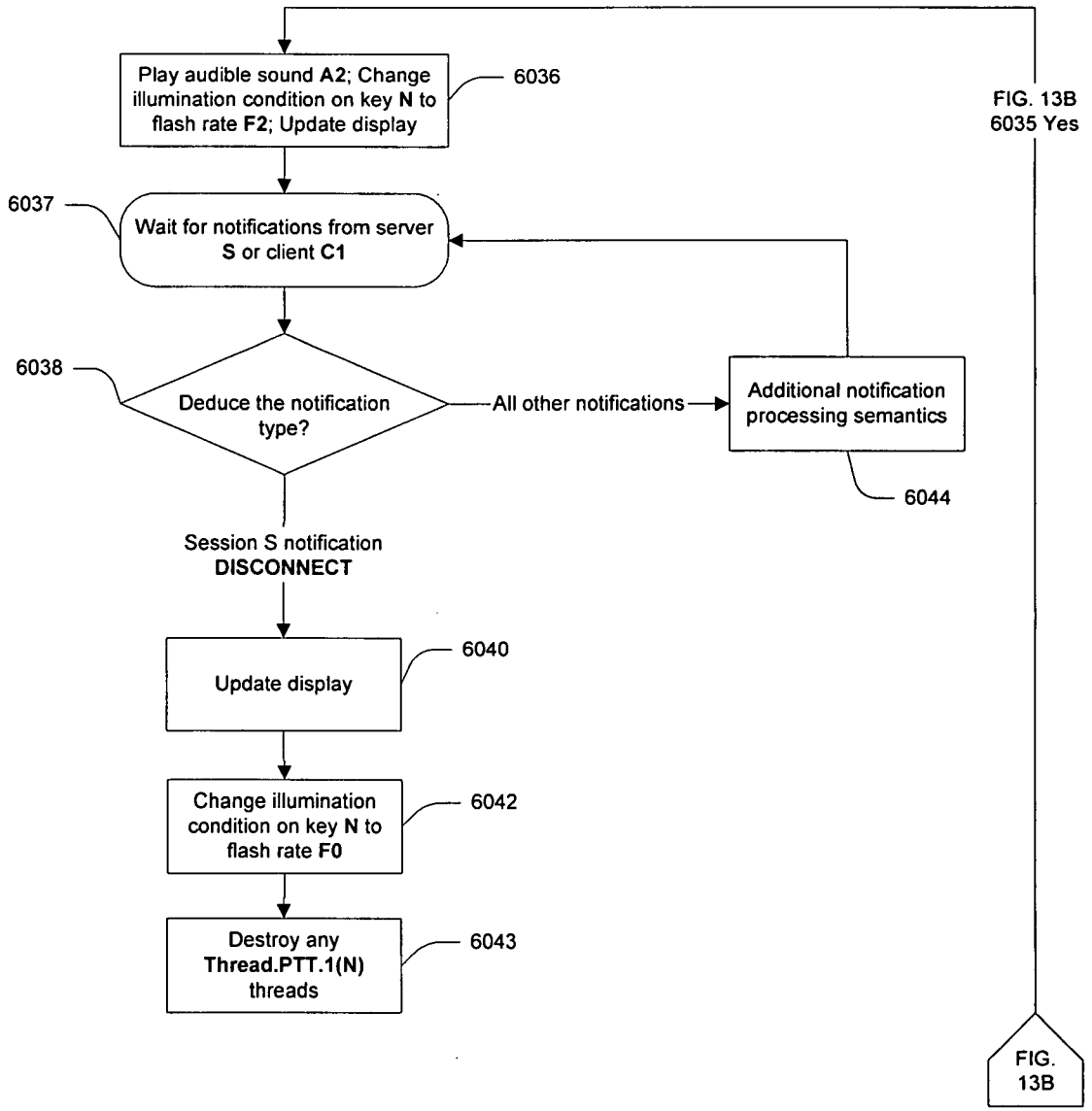

Play audible sound A2; Change illumination condition on key N to flash rate F2; Update display
6036
FIG. 13B
6035 Yes
6037
Wait for notifications from server S or client C1
6038
Deduce the notification type?
All other notifications
Additional notification processing semantics
6044
Session S notification DISCONNECT
Update display
6040
Change illumination condition on key N to flash rate F0
6042
Destroy any Thread.PTT.1(N) threads
6043
FIG. 13B

FIG. 13C

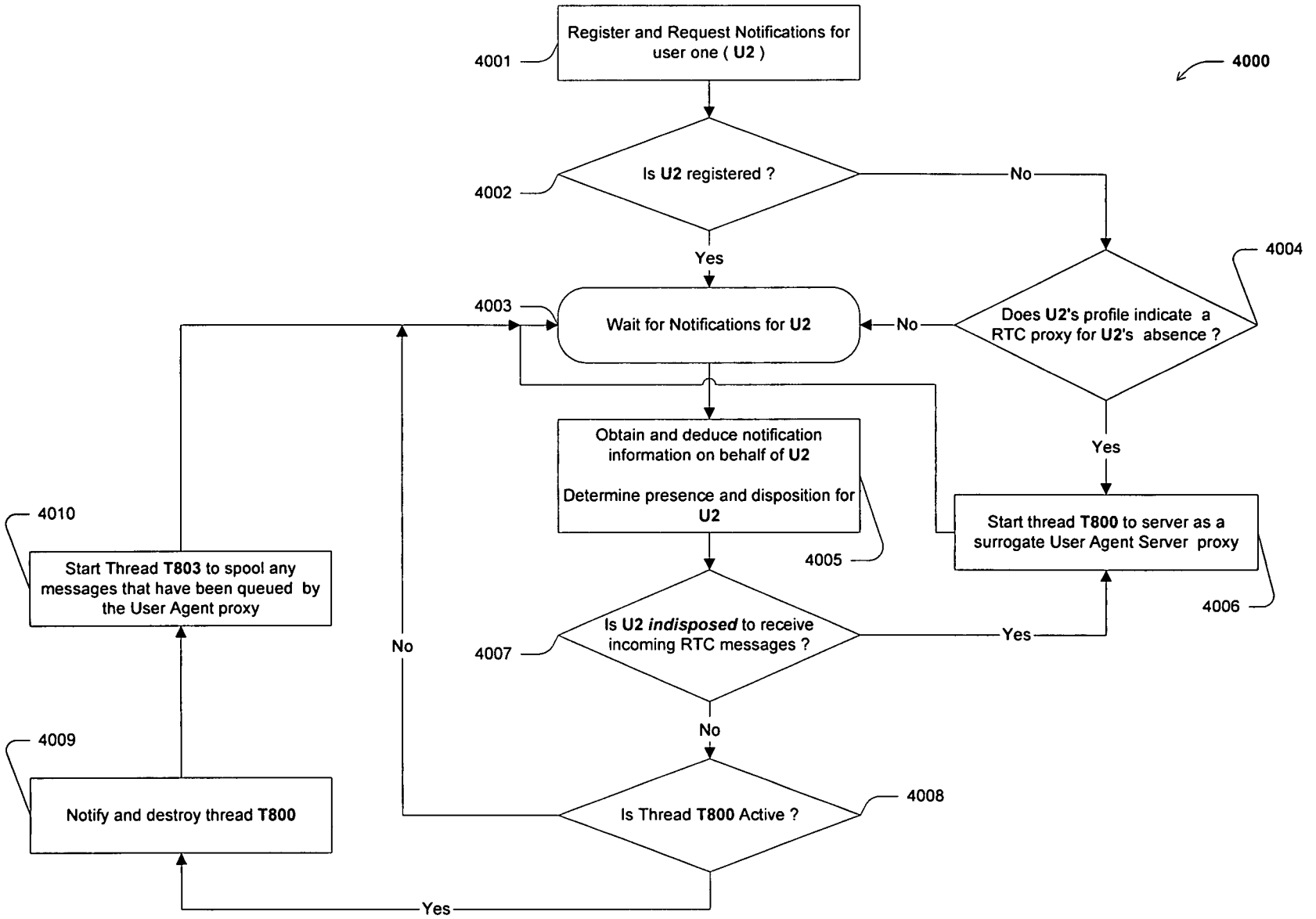

Register and Request Notifications for user one (U2)
4001
4000
Is U2 registered ?
4002
No
Yes
4004
Does U2's profile indicate a RTC proxy for U2's absence ?
4003
Wait for Notifications for U2
No
Yes
Obtain and deduce notification information on behalf of U2
Determine presence and disposition for U2
4005
Start thread T800 to server as a surrogate User Agent Server proxy
4006
4010
Start Thread T803 to spool any messages that have been queued by the User Agent proxy
No
Is U2 indisposed to receive incoming RTC messages ?
4007
Yes
No
4009
Notify and destroy thread T800
Is Thread T800 Active ?
4008
Yes

FIG. 14A

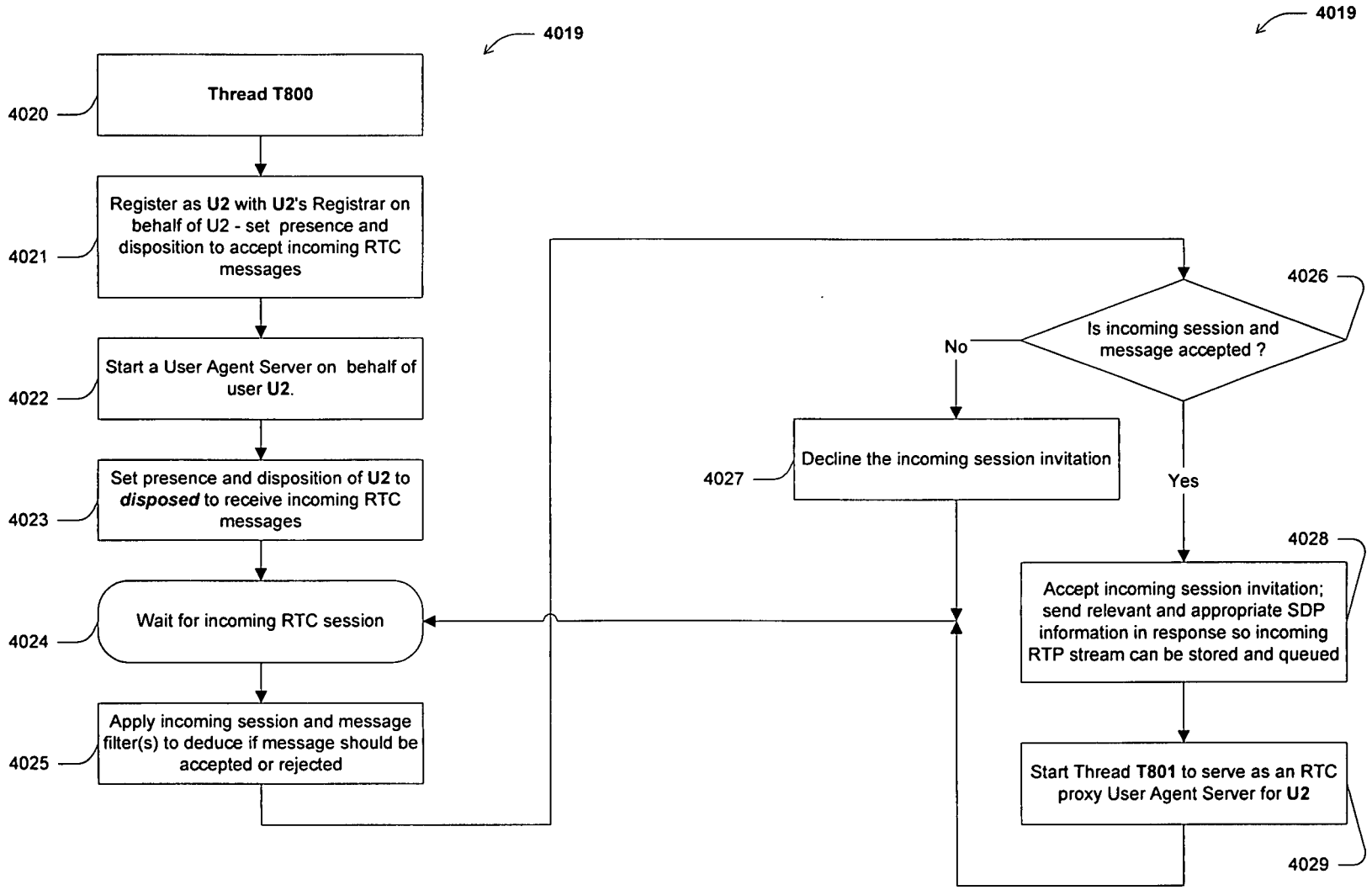

4019
4019
Thread T800
4020
Register as U2 with U2's Registrar on behalf of U2 - set presence and disposition to accept incoming RTC messages
4021
Start a User Agent Server on behalf of user U2.
4022
Set presence and disposition of U2 to disposed to receive incoming RTC messages
4023
Wait for incoming RTC session
4024
Apply incoming session and message filter(s) to deduce if message should be accepted or rejected
4025
4026
Is incoming session and message accepted ?
No
Yes
4027
Decline the incoming session invitation
4028
Accept incoming session invitation; send relevant and appropriate SDP information in response so incoming RTP stream can be stored and queued
Start Thread T801 to serve as an RTC proxy User Agent Server for U2
4029

FIG. 14B

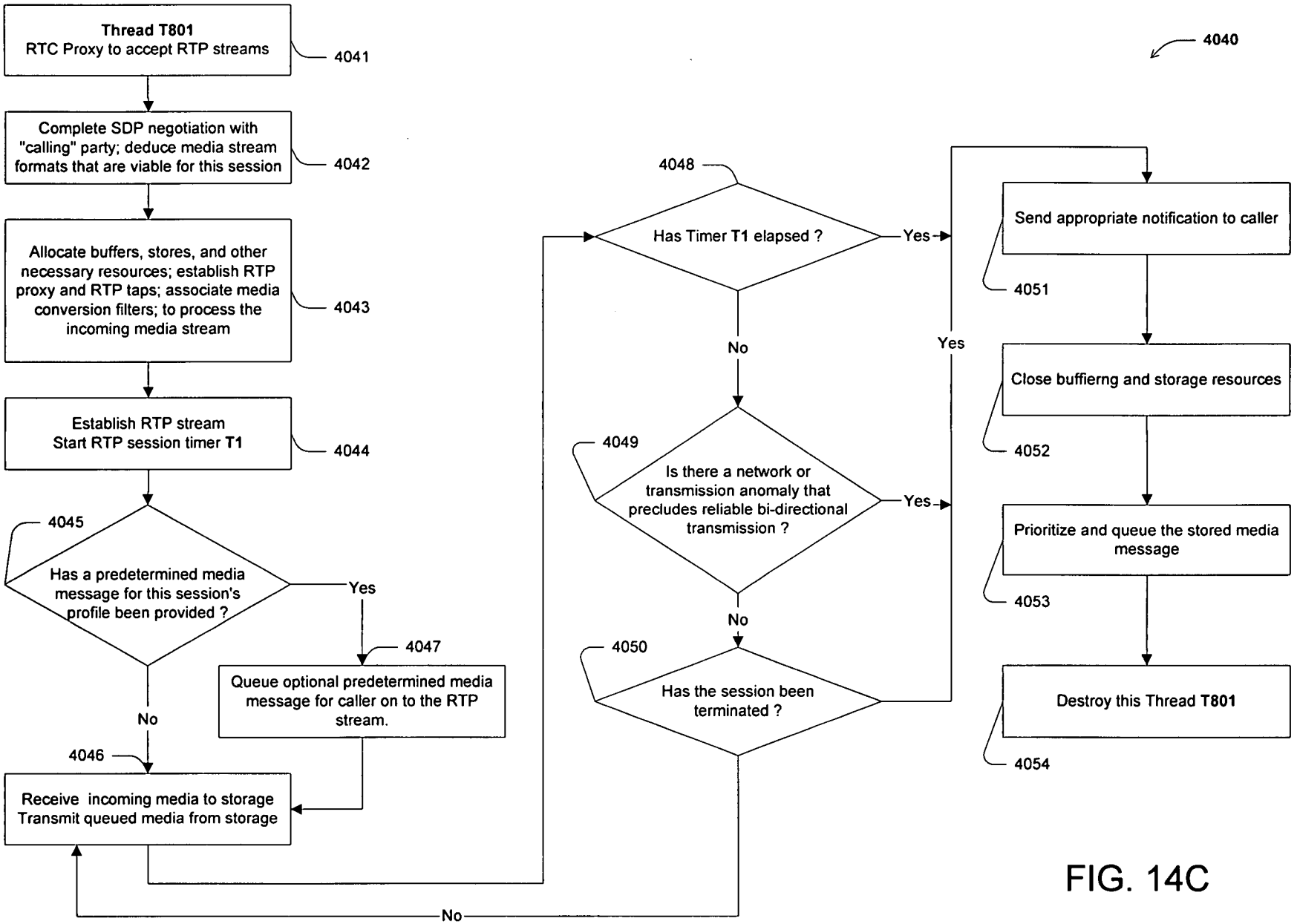

4040
Thread T801
RTC Proxy to accept RTP streams
4041
Complete SDP negotiation with "calling" party; deduce media stream formats that are viable for this session
4042
Allocate buffers, stores, and other necessary resources; establish RTP proxy and RTP taps; associate media conversion filters; to process the incoming media stream
4043
Establish RTP stream
Start RTP session timer T1
4044
4045
Has a predetermined media message for this session's profile been provided ?
Yes
No
4047
Queue optional predetermined media message for caller on to the RTP stream.
4046
Receive incoming media to storage
Transmit queued media from storage
4048
Has Timer T1 elapsed ?
Yes
No
4049
Is there a network or transmission anomaly that precludes reliable bi-directional transmission ?
Yes
No
4050
Has the session been terminated ?
Yes
No
Send appropriate notification to caller
4051
Close buffierng and storage resources
4052
Prioritize and queue the stored media message
4053
Destroy this Thread T801
4054

FIG. 14C

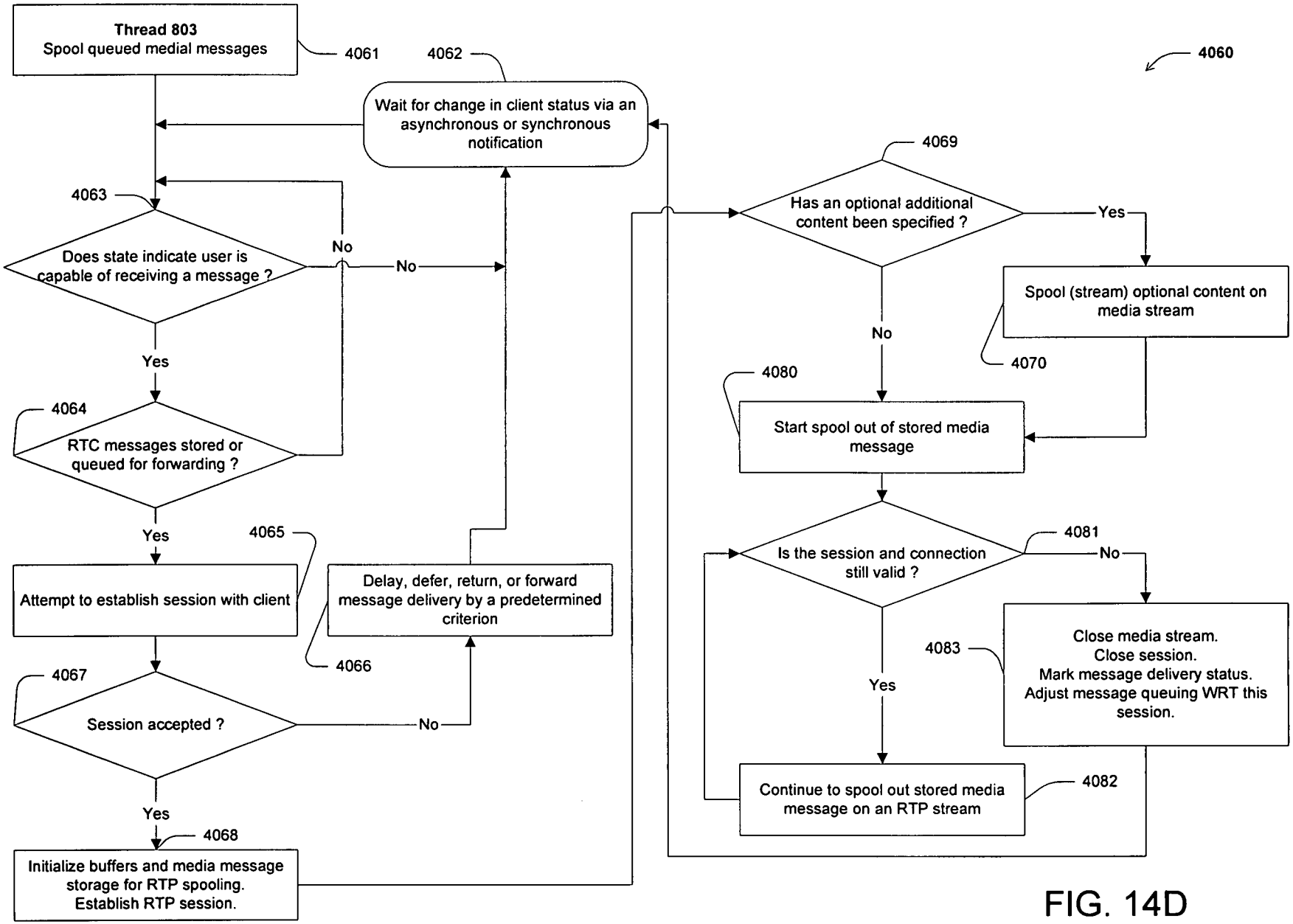

Thread 803
Spool queued medial messages
4061
4062
4060
Wait for change in client status via an asynchronous or synchronous notification
4063
Does state indicate user is capable of receiving a message ?
No
No
Yes
4064
RTC messages stored or queued for forwarding ?
Yes
4065
Attempt to establish session with client
Delay, defer, return, or forward message delivery by a predetermined criterion
4066
4067
Session accepted ?
No
Yes
4068
Initialize buffers and media message storage for RTP spooling. Establish RTP session.
4069
Has an optional additional content been specified ?
Yes
Spool (stream) optional content on media stream
4070
No
4080
Start spool out of stored media message
Is the session and connection still valid ?
4081
No
Yes
4083
Close media stream.
Close session.
Mark message delivery status.
Adjust message queuing WRT this session.
Continue to spool out stored media message on an RTP stream
4082

FIG. 14D

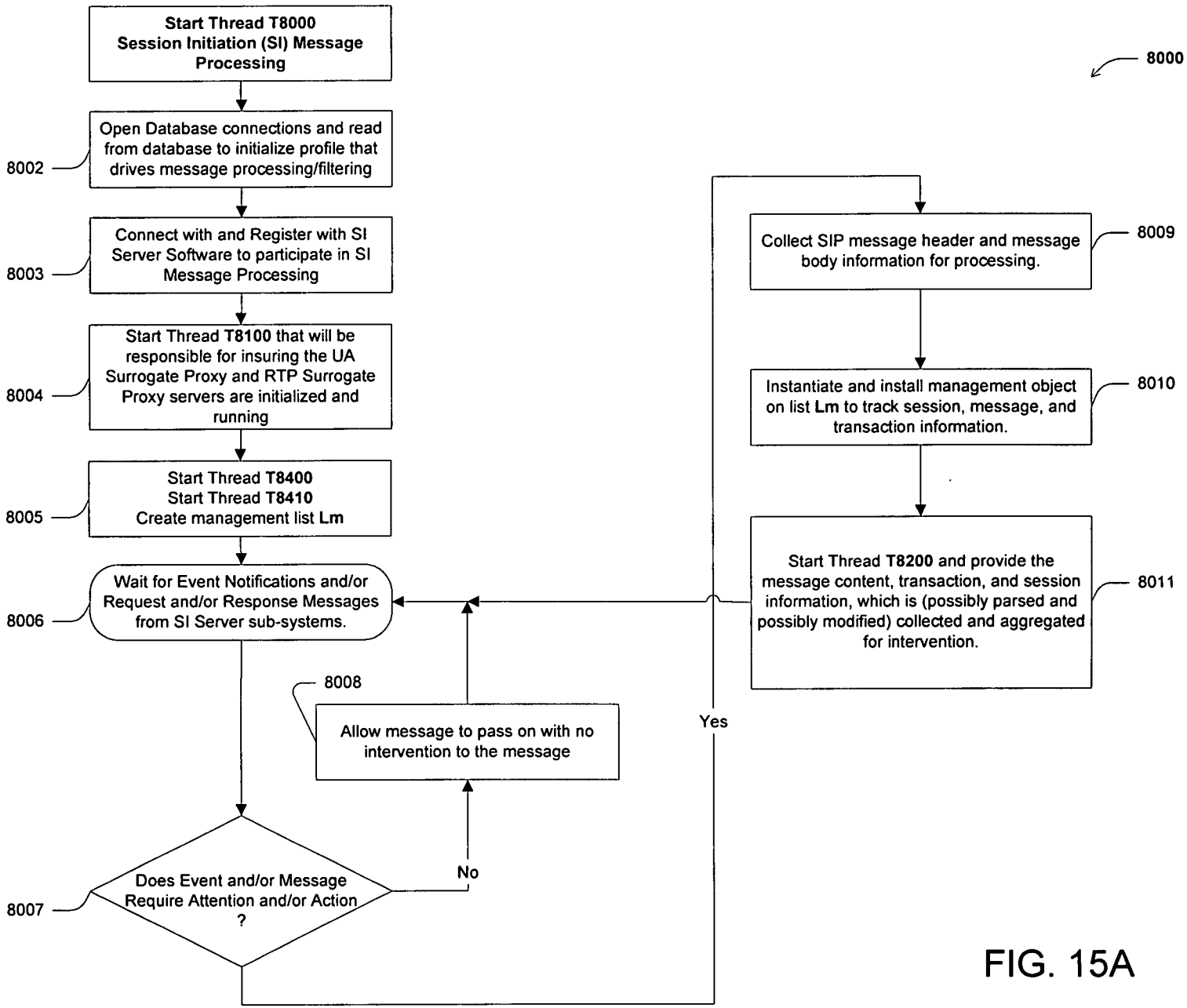

Start Thread T8000
Session Initiation (SI) Message Processing
8000
Open Database connections and read from database to initialize profile that drives message processing/filtering
8002
Connect with and Register with SI Server Software to participate in SI Message Processing
8003
Start Thread T8100 that will be responsible for insuring the UA Surrogate Proxy and RTP Surrogate Proxy servers are initialized and running
8004
Start Thread T8400
Start Thread T8410
Create management list Lm
8005
Wait for Event Notifications and/or Request and/or Response Messages from SI Server sub-systems.
8006
Does Event and/or Message Require Attention and/or Action ?
8007
No
8008
Allow message to pass on with no intervention to the message
Yes
Collect SIP message header and message body information for processing.
8009
Instantiate and install management object on list Lm to track session, message, and transaction information.
8010
Start Thread T8200 and provide the message content, transaction, and session information, which is (possibly parsed and possibly modified) collected and aggregated for intervention.
8011

FIG. 15A

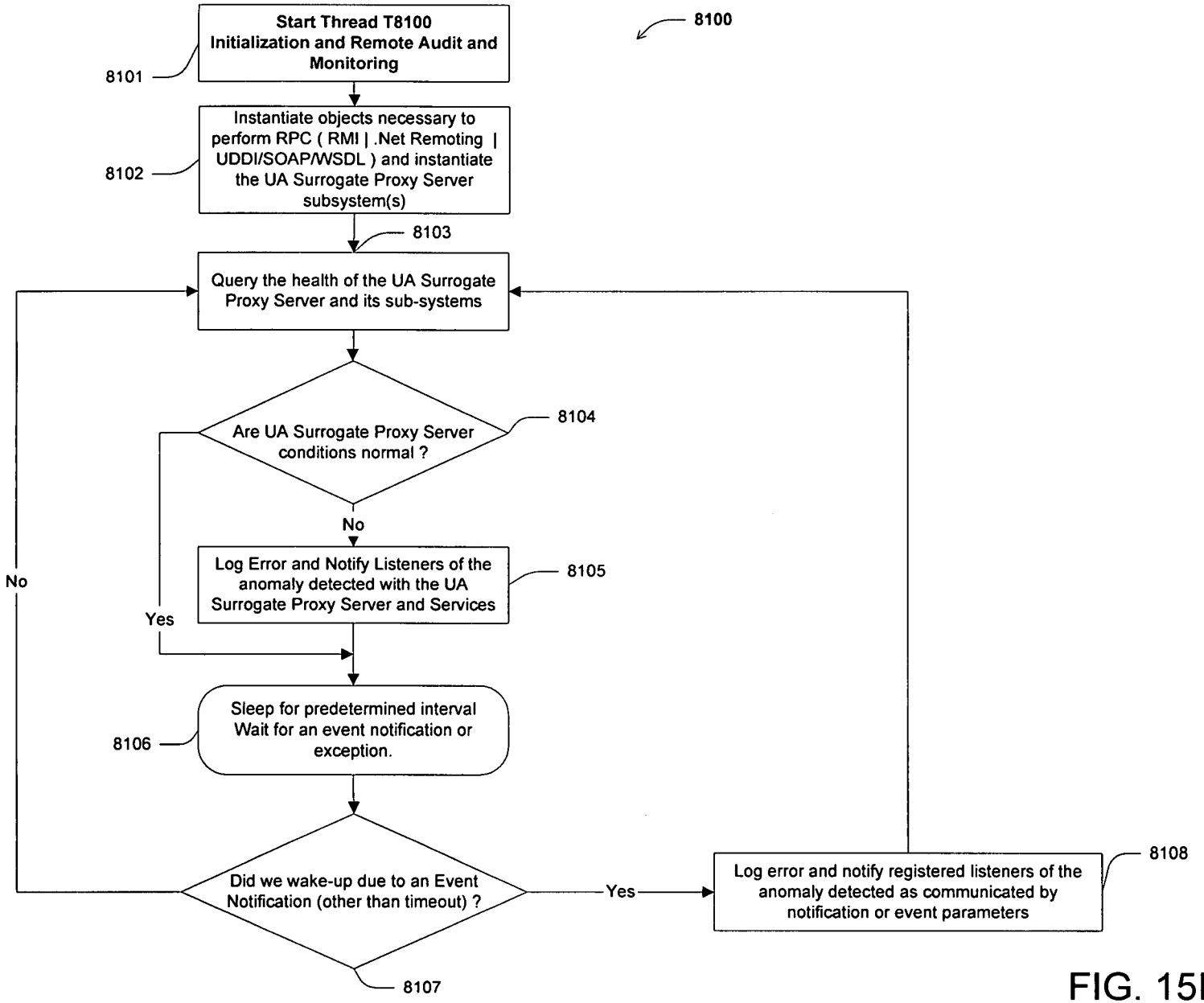

Start Thread T8100
Initialization and Remote Audit and Monitoring
8100
8101
Instantiate objects necessary to perform RPC (RMI | .Net Remoting | UDDI/SOAP/WSDL) and instantiate the UA Surrogate Proxy Server subsystem(s)
8102
8103
Query the health of the UA Surrogate Proxy Server and its sub-systems
Are UA Surrogate Proxy Server conditions normal ?
8104
No
Yes
No
Log Error and Notify Listeners of the anomaly detected with the UA Surrogate Proxy Server and Services
8105
Sleep for predetermined interval Wait for an event notification or exception.
8106
Did we wake-up due to an Event Notification (other than timeout) ?
Yes
8107
Log error and notify registered listeners of the anomaly detected as communicated by notification or event parameters
8108

FIG. 15B

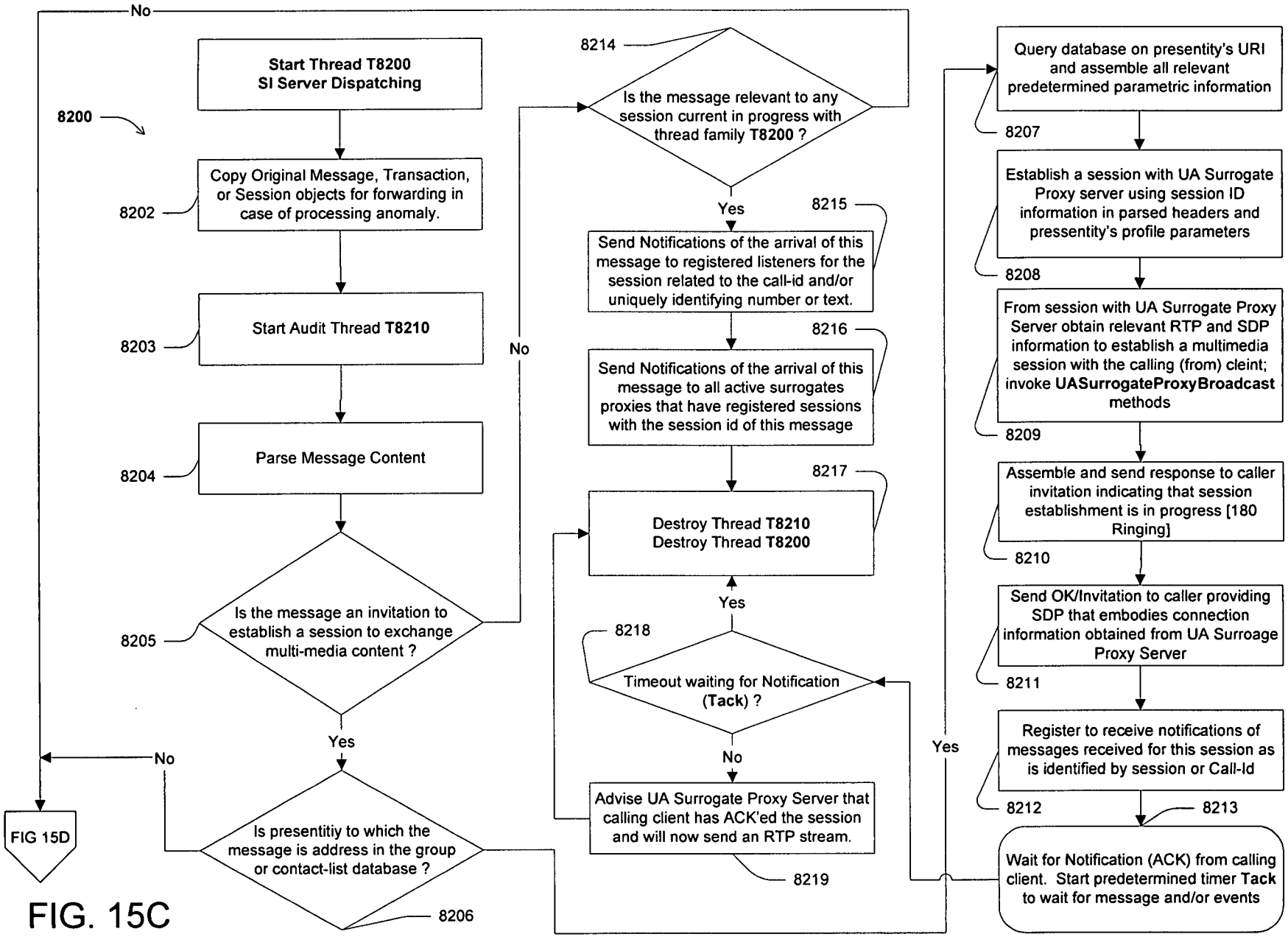

No
Start Thread T8200
SI Server Dispatching
8200
Copy Original Message, Transaction, or Session objects for forwarding in case of processing anomaly.
8202
Start Audit Thread T8210
8203
Parse Message Content
8204
Is the message an invitation to establish a session to exchange multi-media content ?
8205
Yes
No
Is presentitiy to which the message is address in the group or contact-list database ?
8206
FIG 15D
8214
Is the message relevant to any session current in progress with thread family T8200 ?
No
Yes
8215
Send Notifications of the arrival of this message to registered listeners for the session related to the call-id and/or uniquely identifying number or text.
8216
Send Notifications of the arrival of this message to all active surrogates proxies that have registered sessions with the session id of this message
8217
Destroy Thread T8210
Destroy Thread T8200
Yes
8218
Timeout waiting for Notification (Tack) ?
No
Advise UA Surrogate Proxy Server that calling client has ACK'ed the session and will now send an RTP stream.
8219
Yes
Query database on presentity's URI and assemble all relevant predetermined parametric information
8207
Establish a session with UA Surrogate Proxy server using session ID information in parsed headers and pressentity's profile parameters
8208
From session with UA Surrogate Proxy Server obtain relevant RTP and SDP information to establish a multimedia session with the calling (from) cleint; invoke UASurrogateProxyBroadcast methods
8209
Assemble and send response to caller invitation indicating that session establishment is in progress [180 Ringing]
8210
Send OK/Invitation to caller providing SDP that embodies connection information obtained from UA Surroage Proxy Server
8211
Register to receive notifications of messages received for this session as is identified by session or Call-Id
8212
8213
Wait for Notification (ACK) from calling client. Start predetermined timer Tack to wait for message and/or events

FIG. 15C

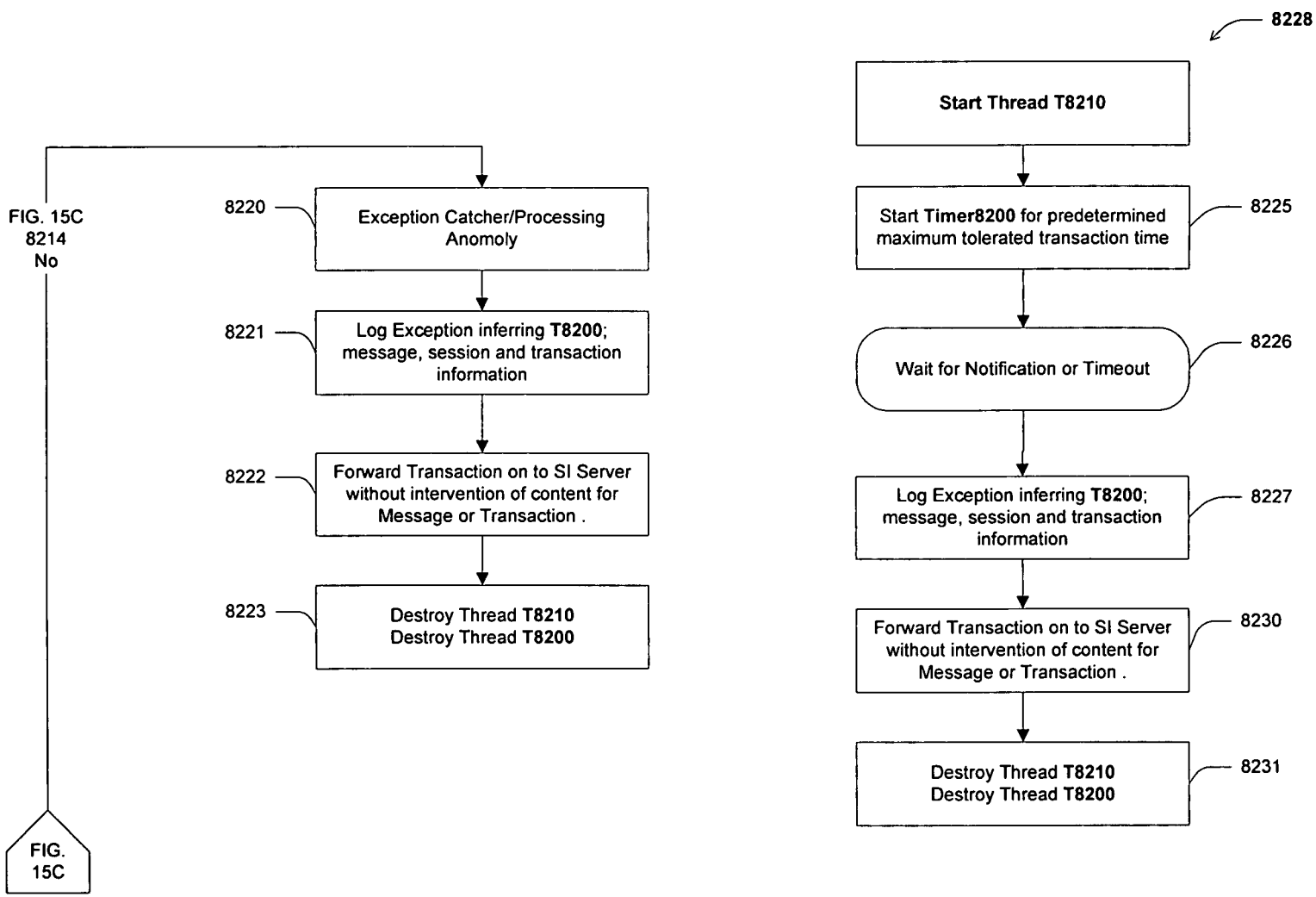


FIG. 15C
8214
No
8220
Exception Catcher/Processing Anomoly
8221
Log Exception inferring T8200; message, session and transaction information
8222
Forward Transaction on to SI Server without intervention of content for Message or Transaction .
8223
Destroy Thread T8210
Destroy Thread T8200
FIG. 15C
8228
Start Thread T8210
Start Timer8200 for predetermined maximum tolerated transaction time
8225
Wait for Notification or Timeout
8226
Log Exception inferring T8200; message, session and transaction information
8227
Forward Transaction on to SI Server without intervention of content for Message or Transaction .
8230
Destroy Thread T8210
Destroy Thread T8200
8231

FIG. 15D

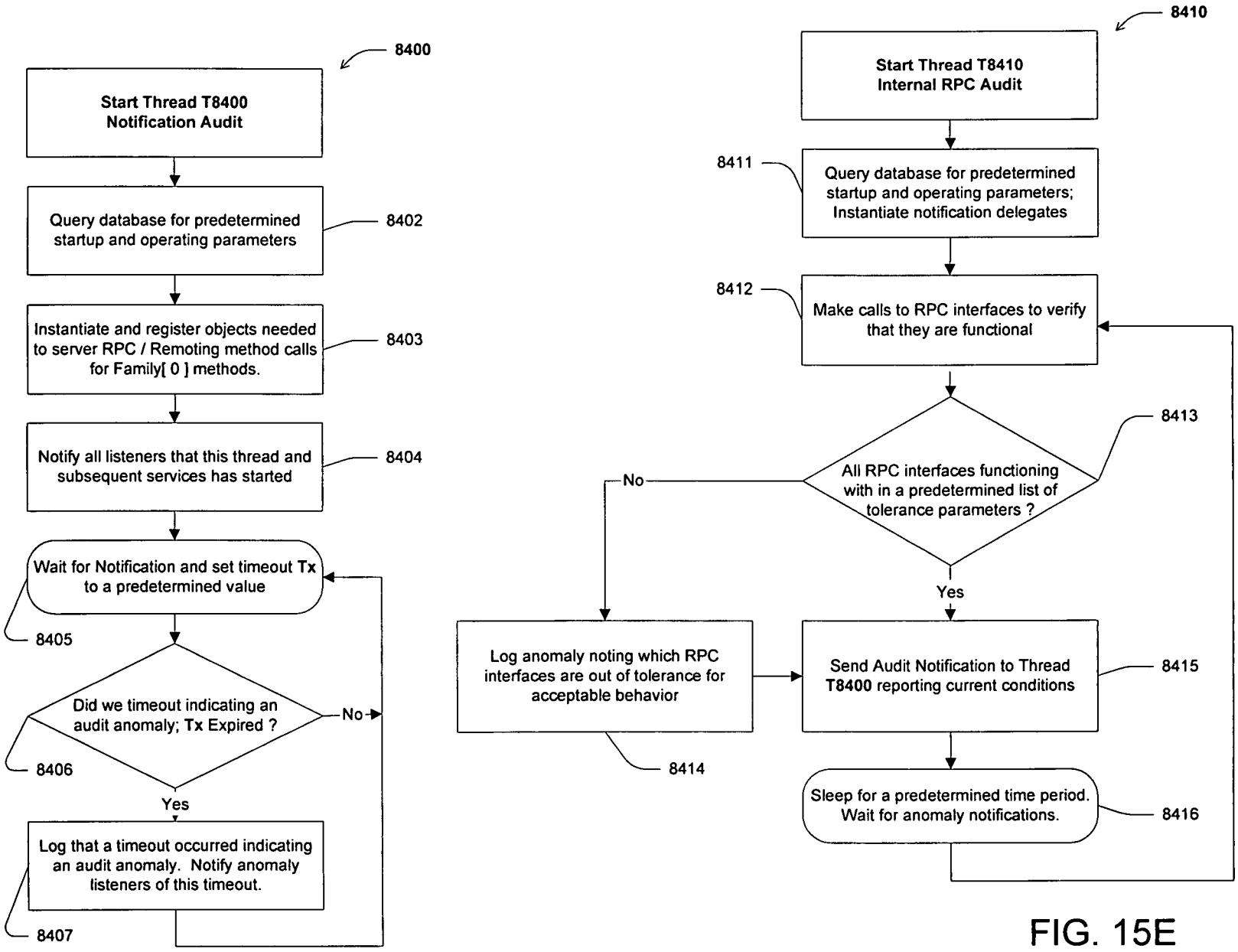

8400
Start Thread T8400
Notification Audit
Query database for predetermined startup and operating parameters
8402
Instantiate and register objects needed to server RPC / Remoting method calls for Family[0] methods.
8403
Notify all listeners that this thread and subsequent services has started
8404
Wait for Notification and set timeout Tx to a predetermined value
8405
Did we timeout indicating an audit anomaly; Tx Expired ?
No
8406
Yes
Log that a timeout occurred indicating an audit anomaly. Notify anomaly listeners of this timeout.
8407
8410
Start Thread T8410
Internal RPC Audit
8411
Query database for predetermined startup and operating parameters; Instantiate notification delegates
8412
Make calls to RPC interfaces to verify that they are functional
8413
All RPC interfaces functioning with in a predetermined list of tolerance parameters ?
No
Yes
Log anomaly noting which RPC interfaces are out of tolerance for acceptable behavior
8414
Send Audit Notification to Thread T8400 reporting current conditions
8415
Sleep for a predetermined time period. Wait for anomaly notifications.
8416

FIG. 15E

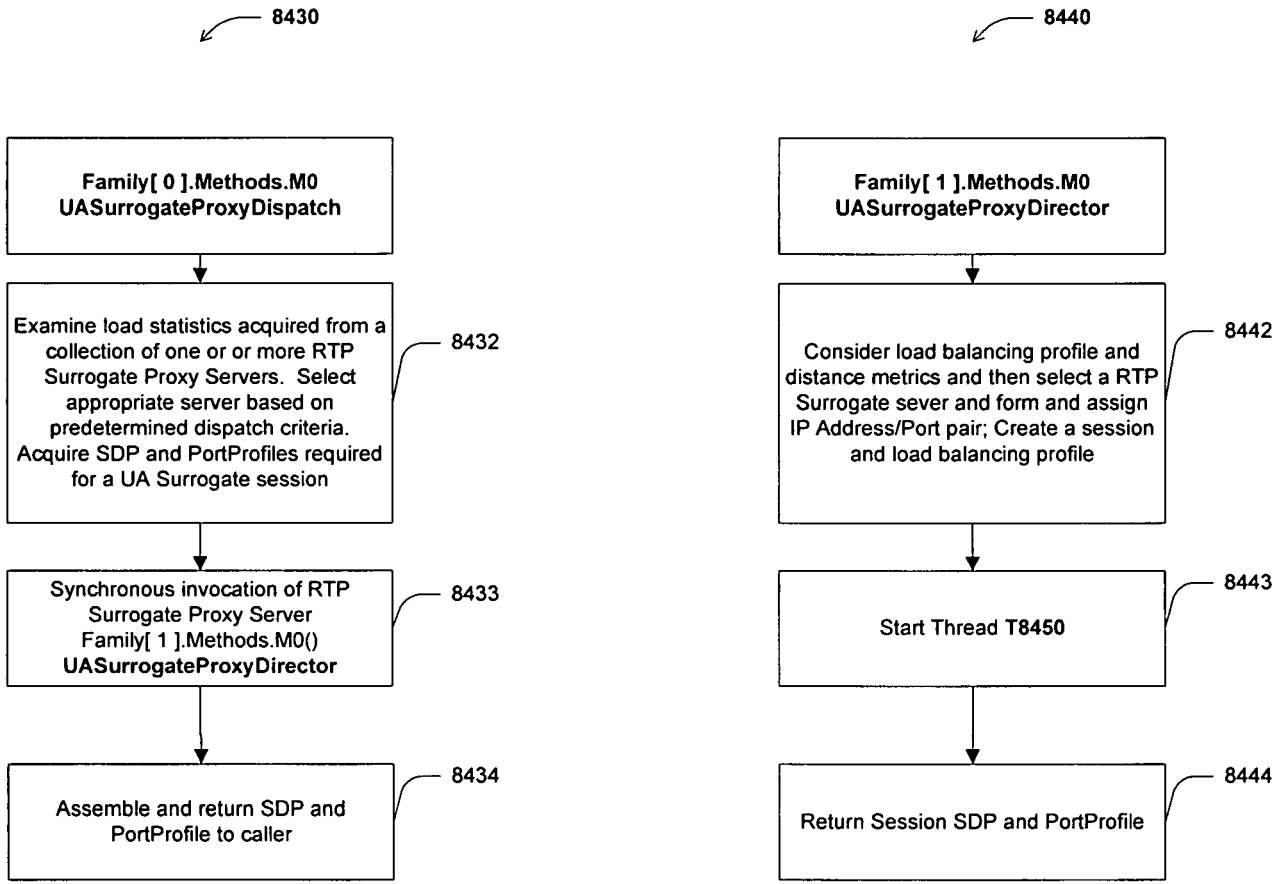

8430
Family[0].Methods.M0
UASurrogateProxyDispatch
Examine load statistics acquired from a collection of one or or more RTP Surrogate Proxy Servers. Select appropriate server based on predetermined dispatch criteria. Acquire SDP and PortProfiles required for a UA Surrogate session
8432
Synchronous invocation of RTP Surrogate Proxy Server Family[1].Methods.M0()
UASurrogateProxyDirector
8433
Assemble and return SDP and PortProfile to caller
8434
8440
Family[1].Methods.M0
UASurrogateProxyDirector
Consider load balancing profile and distance metrics and then select a RTP Surrogate sever and form and assign IP Address/Port pair; Create a session and load balancing profile
8442
Start Thread T8450
8443
Return Session SDP and PortProfile
8444

FIG. 15F

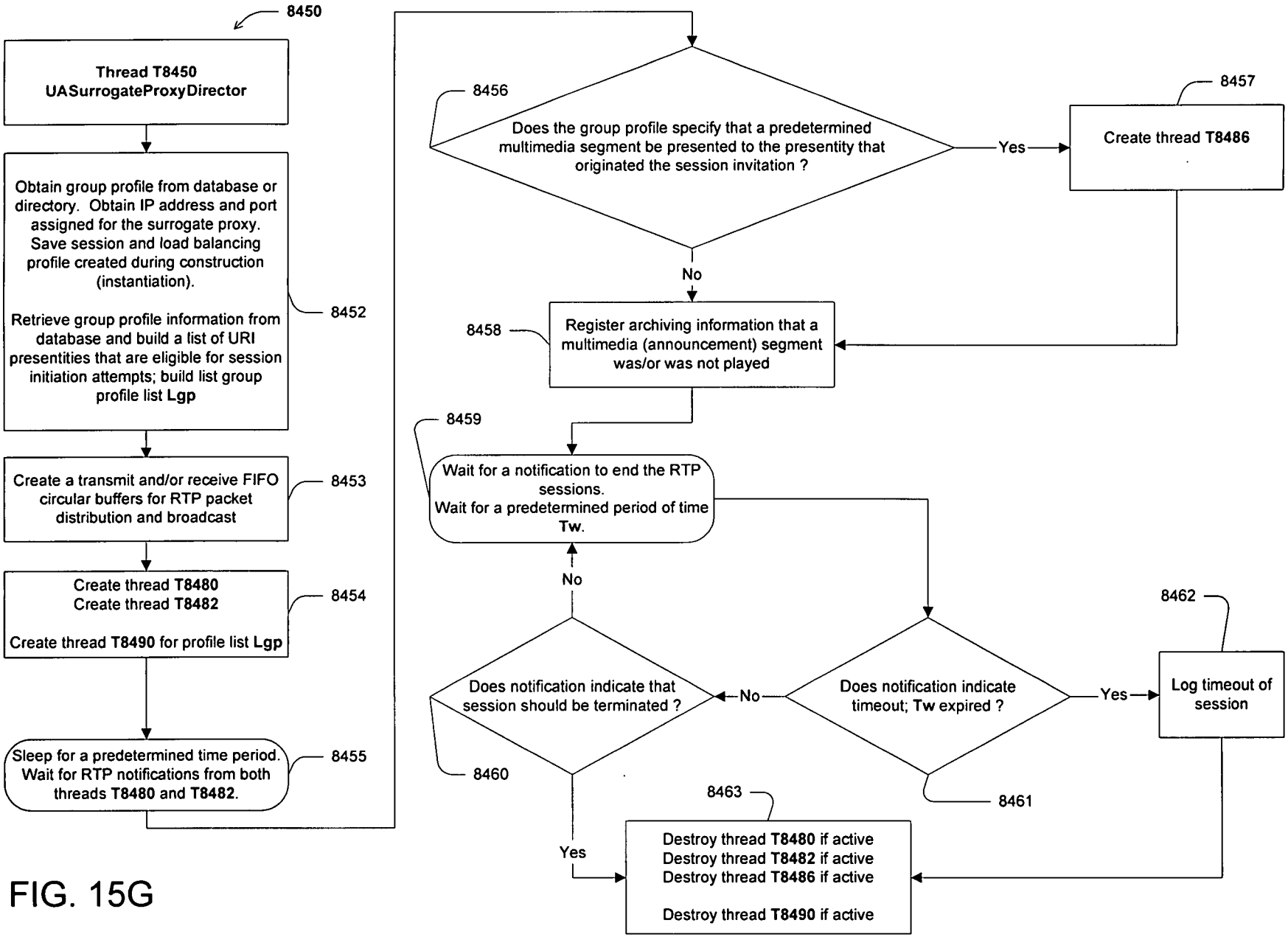

8450
Thread T8450
UASurrogateProxyDirector
Obtain group profile from database or directory. Obtain IP address and port assigned for the surrogate proxy. Save session and load balancing profile created during construction (instantiation).
Retrieve group profile information from database and build a list of URI presentities that are eligible for session initiation attempts; build list group profile list Lgp
8452
Create a transmit and/or receive FIFO circular buffers for RTP packet distribution and broadcast
8453
Create thread T8480
Create thread T8482
Create thread T8490 for profile list Lgp
8454
Sleep for a predetermined time period. Wait for RTP notifications from both threads T8480 and T8482.
8455
8456
Does the group profile specify that a predetermined multimedia segment be presented to the presentity that originated the session invitation ?
Yes
8457
Create thread T8486
No
8458
Register archiving information that a multimedia (announcement) segment was/or was not played
8459
Wait for a notification to end the RTP sessions.
Wait for a predetermined period of time Tw.
No
Does notification indicate that session should be terminated ?
No
Does notification indicate timeout; Tw expired ?
Yes
8462
Log timeout of session
8460
8461
8463
Yes
Destroy thread T8480 if active
Destroy thread T8482 if active
Destroy thread T8486 if active
Destroy thread T8490 if active

FIG. 15G

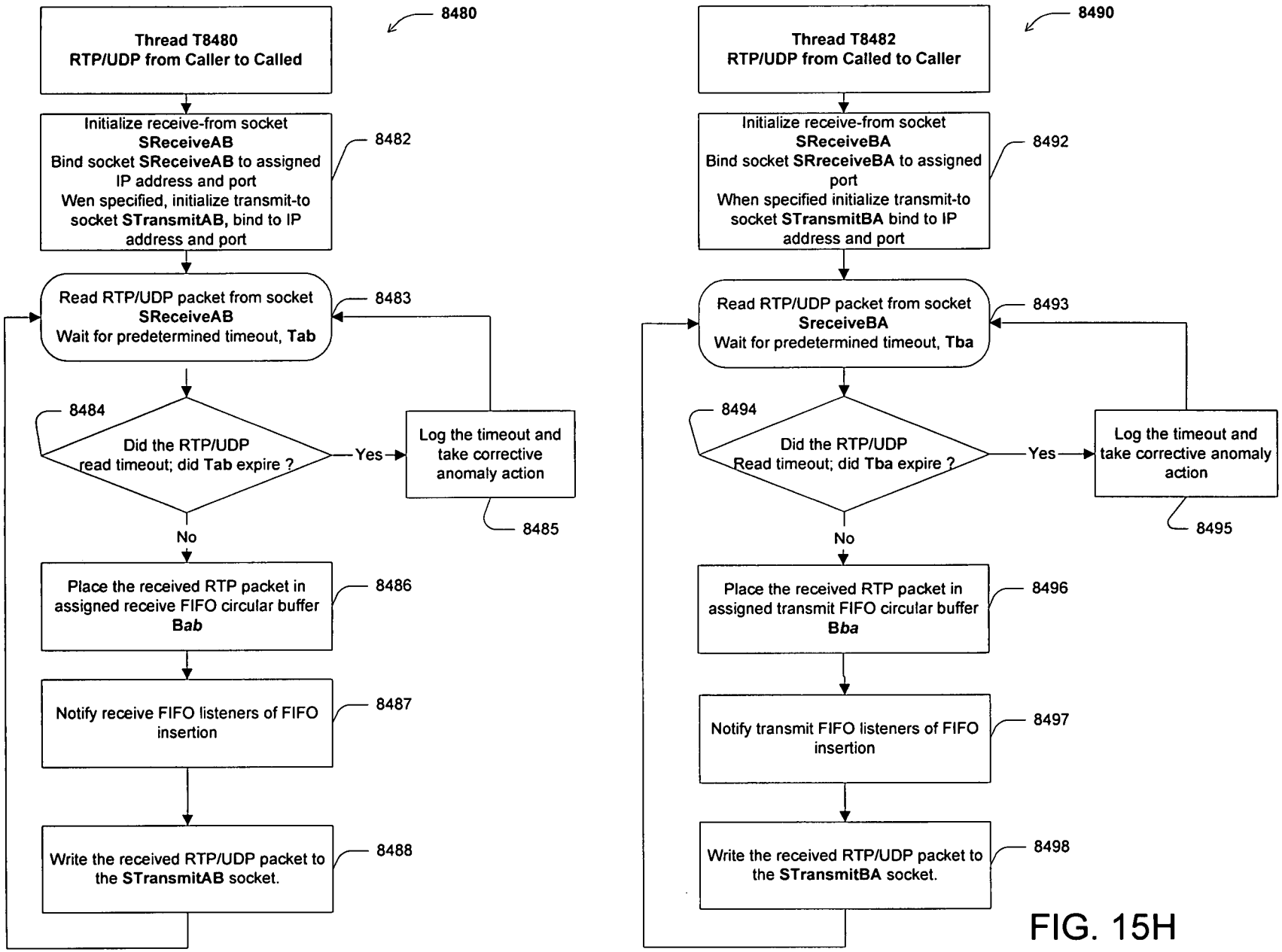

Thread T8480
RTP/UDP from Caller to Called
8480
Initialize receive-from socket
SReceiveAB
Bind socket SReceiveAB to assigned IP address and port
Wen specified, initialize transmit-to socket STransmitAB, bind to IP address and port
8482
Read RTP/UDP packet from socket
SReceiveAB
Wait for predetermined timeout, Tab
8483
8484
Did the RTP/UDP read timeout; did Tab expire ?
Yes
Log the timeout and take corrective anomaly action
8485
No
Place the received RTP packet in assigned receive FIFO circular buffer Bab
8486
Notify receive FIFO listeners of FIFO insertion
8487
Write the received RTP/UDP packet to the STransmitAB socket.
8488
Thread T8482
RTP/UDP from Called to Caller
8490
Initialize receive-from socket
SReceiveBA
Bind socket SRreceiveBA to assigned port
When specified initialize transmit-to socket STransmitBA bind to IP address and port
8492
Read RTP/UDP packet from socket
SreceiveBA
Wait for predetermined timeout, Tba
8493
8494
Did the RTP/UDP Read timeout; did Tba expire ?
Yes
Log the timeout and take corrective anomaly action
8495
No
Place the received RTP packet in assigned transmit FIFO circular buffer Bba
8496
Notify transmit FIFO listeners of FIFO insertion
8497
Write the received RTP/UDP packet to the STransmitBA socket.
8498

FIG. 15H

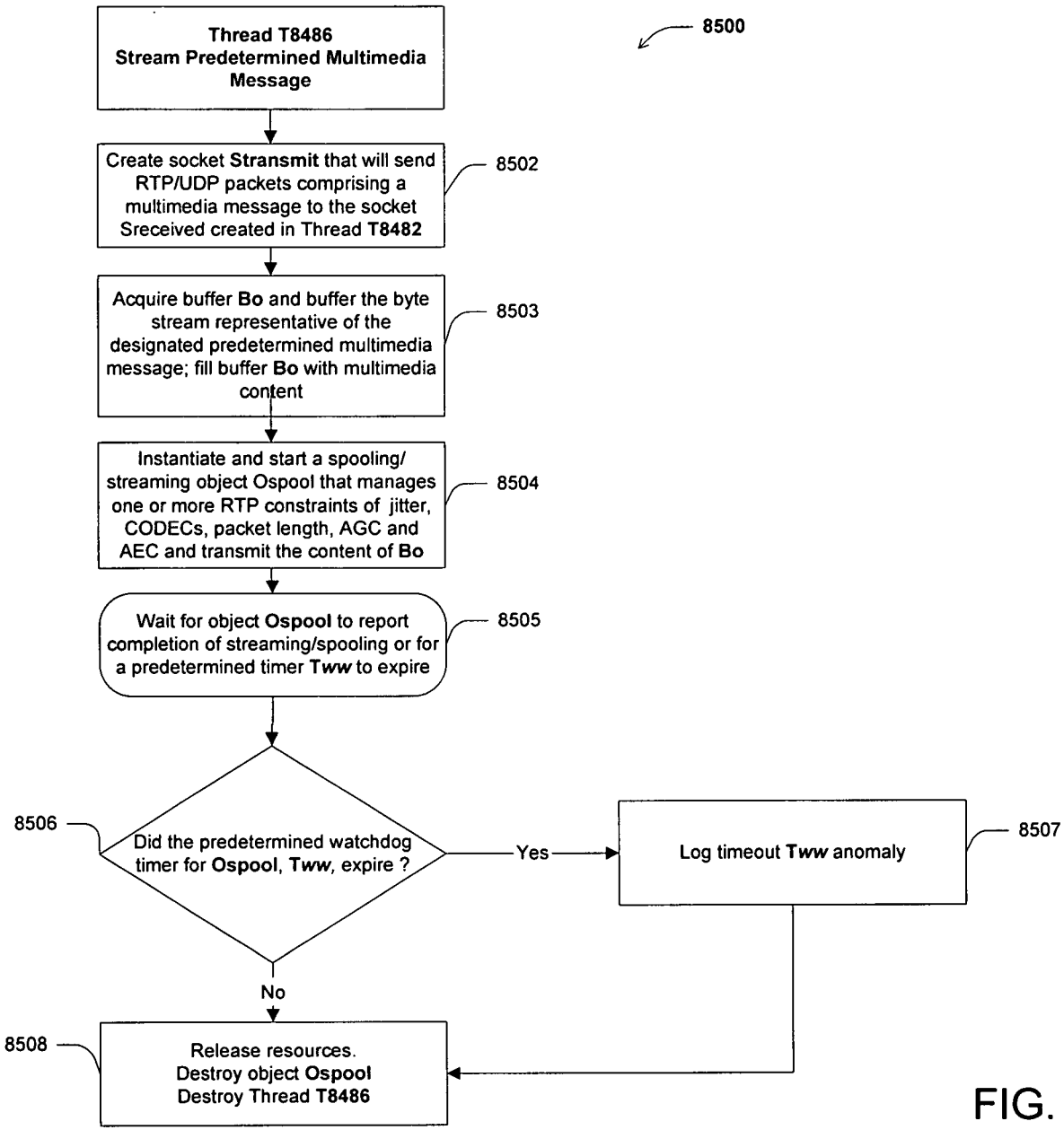

Thread T8486
Stream Predetermined Multimedia Message
8500
Create socket Stransmit that will send RTP/UDP packets comprising a multimedia message to the socket Sreceived created in Thread T8482
8502
Acquire buffer Bo and buffer the byte stream representative of the designated predetermined multimedia message; fill buffer Bo with multimedia content
8503
Instantiate and start a spooling/ streaming object Ospool that manages one or more RTP constraints of jitter, CODECs, packet length, AGC and AEC and transmit the content of Bo
8504
Wait for object Ospool to report completion of streaming/spooling or for a predetermined timer Tww to expire
8505
8506
Did the predetermined watchdog timer for Ospool, Tww, expire ?
Yes
Log timeout Tww anomaly
8507
No
8508
Release resources.
Destroy object Ospool
Destroy Thread T8486

FIG. 15J

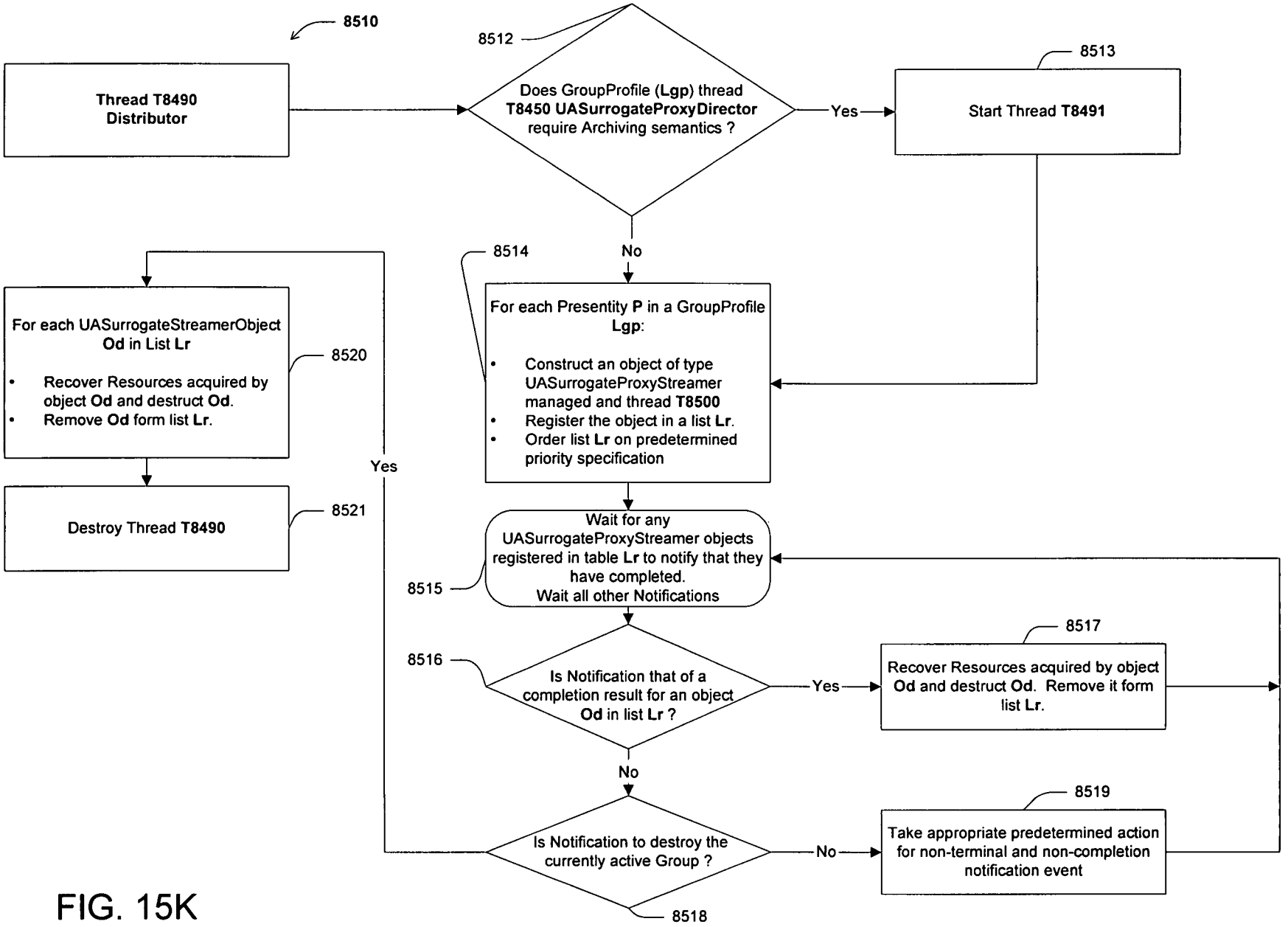

8510
Thread T8490
Distributor
8512
Does GroupProfile (Lgp) thread T8450 UASurrogateProxyDirector require Archiving semantics ?
Yes
8513
Start Thread T8491
No
8514
For each Presentity P in a GroupProfile Lgp:
• Construct an object of type UASurrogateProxyStreamer managed and thread T8500
• Register the object in a list Lr.
• Order list Lr on predetermined priority specification
8515
Wait for any UASurrogateProxyStreamer objects registered in table Lr to notify that they have completed.
Wait all other Notifications
8516
Is Notification that of a completion result for an object Od in list Lr ?
Yes
8517
Recover Resources acquired by object Od and destruct Od. Remove it form list Lr.
No
8518
Is Notification to destroy the currently active Group ?
No
8519
Take appropriate predetermined action for non-terminal and non-completion notification event
Yes
For each UASurrogateStreamerObject Od in List Lr
• Recover Resources acquired by object Od and destruct Od.
• Remove Od form list Lr.
8520
Destroy Thread T8490
8521

FIG. 15K

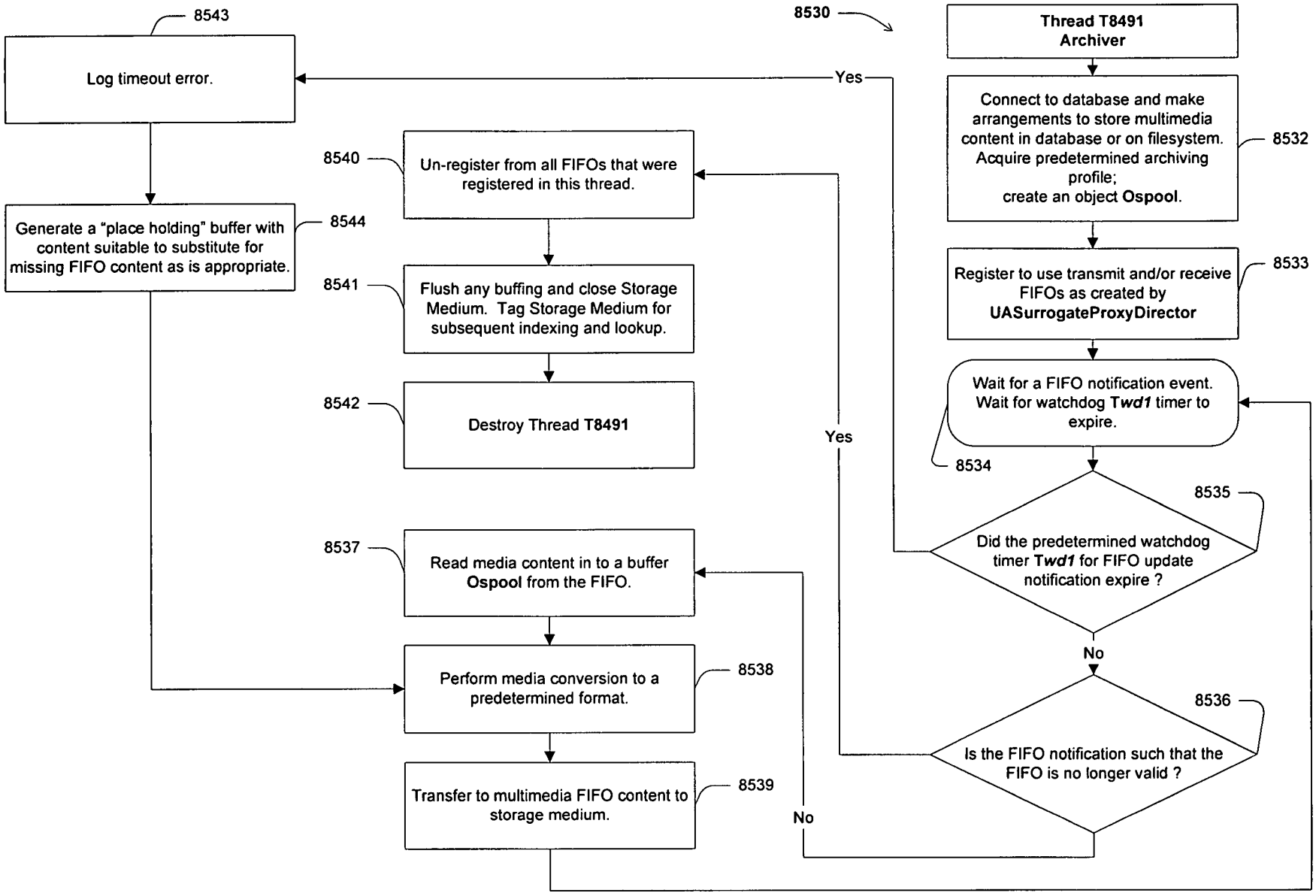

8530
Thread T8491
Archiver
Connect to database and make arrangements to store multimedia content in database or on filesystem. Acquire predetermined archiving profile; create an object Ospool.
8532
Register to use transmit and/or receive FIFOs as created by UASurrogateProxyDirector
8533
Wait for a FIFO notification event. Wait for watchdog Twd1 timer to expire.
8534
Did the predetermined watchdog timer Twd1 for FIFO update notification expire ?
8535
Yes
No
Is the FIFO notification such that the FIFO is no longer valid ?
8536
Yes
No
Log timeout error.
8543
Un-register from all FIFOs that were registered in this thread.
8540
Flush any buffing and close Storage Medium. Tag Storage Medium for subsequent indexing and lookup.
8541
Destroy Thread T8491
8542
Generate a "place holding" buffer with content suitable to substitute for missing FIFO content as is appropriate.
8544
Read media content in to a buffer Ospool from the FIFO.
8537
Perform media conversion to a predetermined format.
8538
Transfer to multimedia FIFO content to storage medium.
8539

FIG. 15L

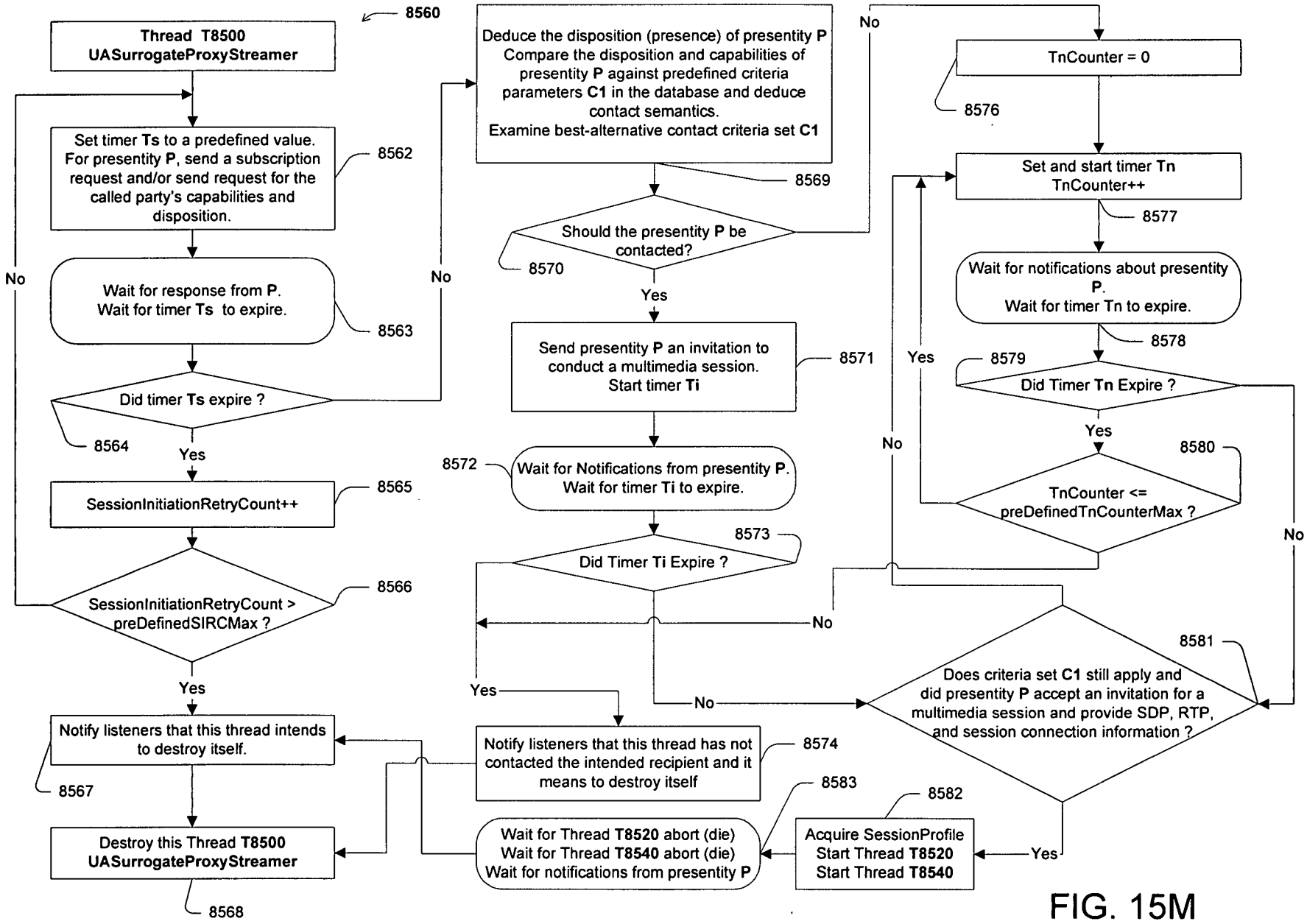

8560
Thread T8500
UASurrogateProxyStreamer
Set timer Ts to a predefined value.
For presentity P, send a subscription request and/or send request for the called party's capabilities and disposition.
8562
Wait for response from P.
Wait for timer Ts to expire.
8563
Did timer Ts expire ?
8564
No
Yes
SessionInitiationRetryCount++
8565
SessionInitiationRetryCount > preDefinedSIRCMax ?
8566
Yes
No
Notify listeners that this thread intends to destroy itself.
8567
Destroy this Thread T8500
UASurrogateProxyStreamer
8568
Deduce the disposition (presence) of presentity P
Compare the disposition and capabilities of presentity P against predefined criteria parameters C1 in the database and deduce contact semantics.
Examine best-alternative contact criteria set C1
8569
Should the presentity P be contacted?
8570
No
Yes
Send presentity P an invitation to conduct a multimedia session.
Start timer Ti
8571
8572
Wait for Notifications from presentity P.
Wait for timer Ti to expire.
8573
Did Timer Ti Expire ?
No
Yes
Notify listeners that this thread has not contacted the intended recipient and it means to destroy itself
8574
Wait for Thread T8520 abort (die)
Wait for Thread T8540 abort (die)
Wait for notifications from presentity P
8583
Acquire SessionProfile
Start Thread T8520
Start Thread T8540
8582
Yes
TnCounter = 0
8576
Set and start timer Tn
TnCounter++
8577
Wait for notifications about presentity P.
Wait for timer Tn to expire.
8578
8579
Did Timer Tn Expire ?
Yes
No
8580
TnCounter <= preDefinedTnCounterMax ?
Yes
No
8581
Does criteria set C1 still apply and did presentity P accept an invitation for a multimedia session and provide SDP, RTP, and session connection information ?
No
Yes

FIG. 15M

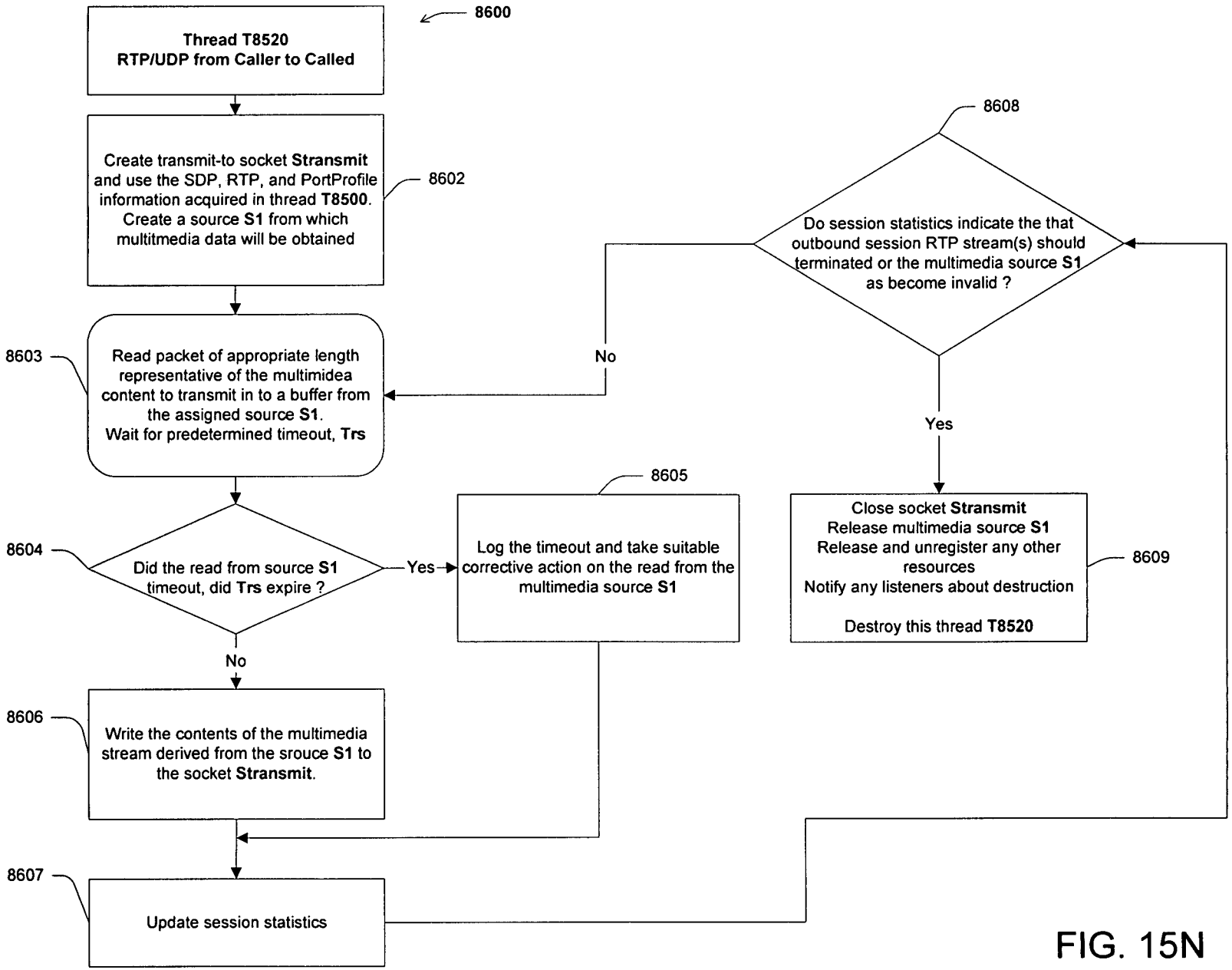

8600
Thread T8520
RTP/UDP from Caller to Called
Create transmit-to socket Stransmit and use the SDP, RTP, and PortProfile information acquired in thread T8500. Create a source S1 from which multitmedia data will be obtained
8602
8603
Read packet of appropriate length representative of the multimidea content to transmit in to a buffer from the assigned source S1. Wait for predetermined timeout, Trs
8604
Did the read from source S1 timeout, did Trs expire ?
Yes
8605
Log the timeout and take suitable corrective action on the read from the multimedia source S1
No
8606
Write the contents of the multimedia stream derived from the srouce S1 to the socket Stransmit.
8607
Update session statistics
8608
Do session statistics indicate the that outbound session RTP stream(s) should terminated or the multimedia source S1 as become invalid ?
No
Yes
Close socket Stransmit
Release multimedia source S1
Release and unregister any other resources
Notify any listeners about destruction
Destroy this thread T8520
8609

FIG. 15N